Investigating Dynamic Routing in Tree-Structured LSTM for Sentiment Analysis

Jin Wang[1], Liang-Chih Yu[2,4], K. Robert Lai[3,4] and Xuejie Zhang[1]
[1]School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China
[2]Department of Information Management, Yuan Ze University, Taiwan
[3]Department of Computer Science & Engineering, Yuan Ze University, Taiwan
[4]Innovation Center for Big Data and Digital Convergence Yuan Ze University, Taiwan
Contact: lcyu@saturn.yzu.edu.tw, xjzhang@ynu.edu.cn

Abstract

Deep neural network models such as long short-term memory (LSTM) and tree-LSTM have been proven to be effective for sentiment analysis. However, sequential LSTM is a bias model wherein the words in the tail of a sentence are more heavily emphasized than those in the header for building sentence representations. Even tree-LSTM, with useful structural information, could not avoid the bias problem because the root node will be dominant and the nodes in the bottom of the parse tree will be less emphasized even though they may contain salient information. To overcome the bias problem, this study proposes a capsule tree-LSTM model, introducing a dynamic routing algorithm as an aggregation layer to build sentence representation by assigning different weights to nodes according to their contributions to prediction. Experiments on Stanford Sentiment Treebank (SST) for sentiment classification and EmoBank for regression show that the proposed method improved the performance of tree-LSTM and other neural network models. In addition, the deeper the tree structure, the bigger the improvement.

1 Introduction

In sentiment analysis, word embeddings (Mikolov et al., 2013a; Mikolov et al., 2013b; Pennington et al., 2014) and sentiment embeddings (Tang et al., 2016; Yu et al., 2018a; Yu et al., 2018b) have become a fundamental component to build deep neural networks such as convolutional neural networks (CNN) (Kalchbrenner et al., 2014; Kim, 2014), recurrent neural networks (RNN) (Graves, 2012; Irsoy and Cardie, 2014), gated recurrent unit (GRU) (Cho et al., 2014), and long short-term memory (LSTM) (Tai et al., 2015; Wang et al., 2015). Given a variable-length text, one challenge of using these neural networks is to compose individual word vectors into sentence vectors with the same length (Iyyer et al., 2015; Joulin et al., 2016; Bojanowski et al., 2016).

The sequential neural networks such as RNN, GRU, and LSTM are commonly used due to their ability to capture long-distance dependency in sequential texts. However, these methods belong to the biased model, where the words in the tail of a sentence are more heavily emphasized than those in the header for building sentence representations. As shown in Fig. 1(a), the priority for each word vector will be "*fantastic* ≻ *really* ≻ *is* ≻ *story* ≻ *this*". This prioritization seems satisfactory for this sentence, but note that the key components could appear anywhere in the sentence rather than necessarily at the end.

To improve the abovementioned sequential models, Tai et al. (2015) and Huang et al. (2017) proposed a tree-LSTM model to introduce useful structural information from sentence parse trees. However, the tree-LSTM also heavily emphasizes the root node in the tree to build sentence representations. That is, words that are closed to the root will be given higher priority than words that are far away from the root. As shown in Fig. 1(b), the priority of word vectors would be "*this* = *story* = *is* ≻ *really* = *fantastic*". This example shows that the tree-LSTM still could not avoid the bias problem because the nodes (e.g., *fantastic*) that contribute more to the prediction but lie in the leaf node at the bottom of the parse tree will be less emphasized.

To overcome the bias problem that may arise in the tree-LSTM, this study proposes a capsule tree-LSTM model. In spired by recent promising work of capsule network (Sabour et al., 2017), the

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3432–3437,
Hong Kong, China, November 3–7, 2019.

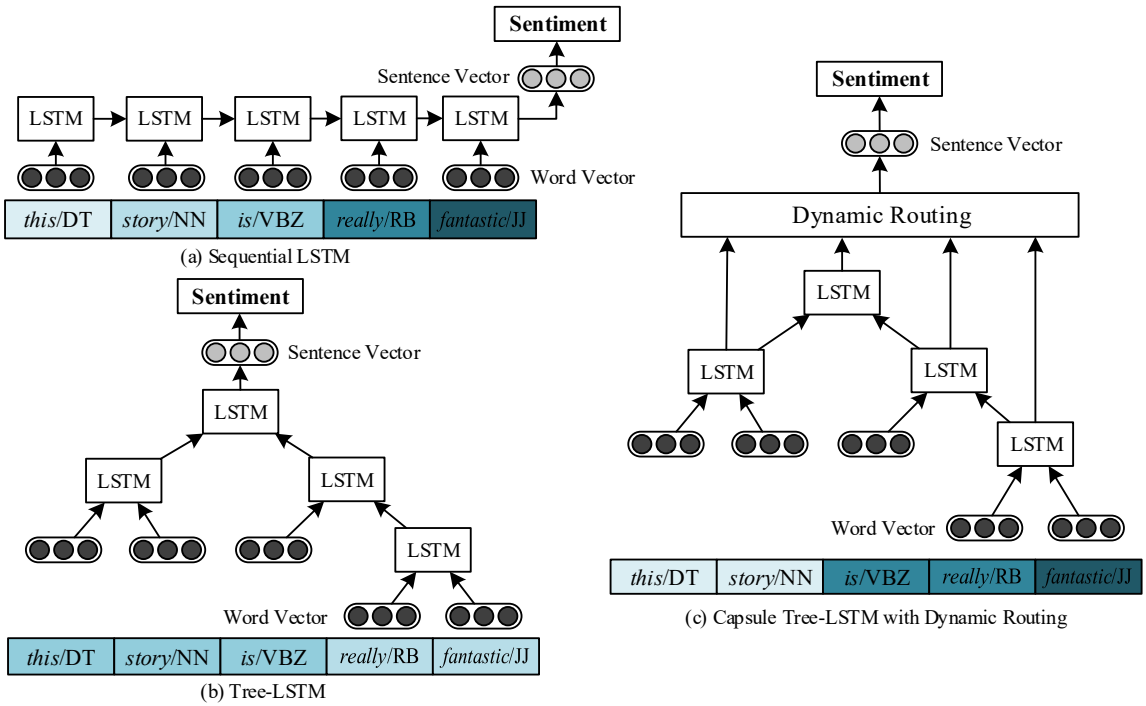

Figure 1: Illustrative examples of different LSTM models for sentiment analysis. A deeper color indicates more weight is assigned to the word according to its contribution to the prediction result.

proposed method introduces a dynamic routing algorithm to consider all non-leaf nodes to build sentence vectors, instead of using the root alone in the tree-LSTM. In addition, different nodes will receive different weights according to their contributions to the prediction task. Unlike self-attention (Lin et al., 2017; Yang et al., 2016), which applies a fixed policy without considering the state of the final sentence vectors, the task of assigning weights in the proposed model is considered to be a routing issue to iteratively determine how much information can be passed from non-leaf nodes in the tree to the vector presentation of the sentence, according to the state of final output. For example, in the aforementioned example text, it would be useful for the model to emphasize *fantastic* that contains the most salient information, even when the word lies at the bottom of the parser tree. Based on the dynamic routing algorithm, the priority of the word vector in the proposed model would be "*fantastic* ≻ *really*=*is* ≻ *this*=*story*". The proposed method is evaluated through both sentiment classification and regression tasks to determine whether dynamic routing can improve the performance of the tree-LSTM and other neural network models.

The rest of this paper is organized as follow. Section 2 describes the proposed capsule tree-LSTM model with dynamic routing. Section 3 summarizes the evaluation results. Conclusions are presented in Section 4.

2 Capsule Tree-LSTM Model

Figure 1(c) shows the framework of the proposed model. First, the given sentence is parsed as a tree-structured topology. The vector representation of this sentence is then generated by composing the word vectors of all non-leaf nodes in the tree according to their weights learned by the dynamic routing algorithm. Finally, the composed sentence vector is used for sentiment prediction.

2.1 Tree-structured LSTM

Given a binary parser tree, the leaf nodes are words and the non-leaf nodes are multi-word phrases. Let $C(j)$ denotes the set of left and right child nodes of a non-leaf node j. Different from the sequential LSTM, the hidden state $\tilde{h}$ of the non-leaf node j is the composition of its left and right child nodes, defined as

$$\tilde{h} \quad \begin{bmatrix} \mathbf{h}_{t-1}^{left} \\ \mathbf{h}_{t-1}^{right} \end{bmatrix} + b_c \qquad (1)$$

where $\mathbf{h}_{t-1}^{left}$ and $\mathbf{h}_{t-1}^{right}$ respectively denote the hidden states of left and right child nodes,

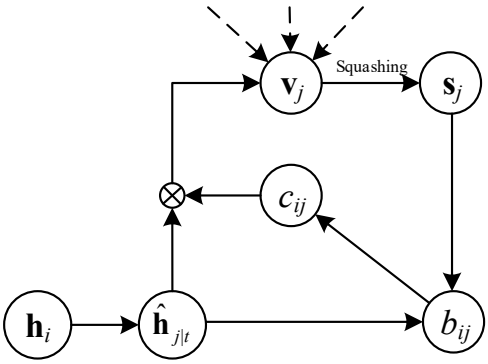

Figure 2: Detailed dynamic routing process

$W_c \in \Re^{d\times 2d}$ is a composition matrix, and b_c is a bias. The tree-LSTM transition equations of node j are defined as

- Gates

$$\begin{aligned} i_t^j &= \sigma(W_{xi}x_t + W_{hi}\tilde{]} \\ f_t^k &= \sigma(W_{xf}x_t + W_{hf}\mathbf{h}_{t-1}^k + b_f) \\ o_t^j &= \sigma(W_{xo}x_t + W_{ho}\tilde{]} \end{aligned} \tag{2}$$

- Input transform

$$c\_in_t = \tanh(W_{xc}x_t + W_{hc}\mathbf{h}_{t-1} + b_{c\_in}) \tag{3}$$

- Memory update

$$\begin{aligned} c_t^j &= i_t^j \otimes c\_in_t^j + \sum_{k\in C(j)} f_t^k \otimes c_{t-1}^k \\ \mathbf{h}_t^j &= o_t^j \otimes \tanh(c_t^j) \end{aligned} \tag{4}$$

where i_t, f_t, o_t, and c_t respectively denote the input gate, forget gate, output gate, and memory cell of node j, x_t denotes the input word vector at the time step t, σ denotes the logistic sigmoid function, W and b respectively denote the weights and bias, and $\otimes$ denotes element-wise multiplication. To integrate the sequence information in the output layer, the order of non-leaf hidden states to form the input matrix of dynamic routing layer is a key consideration. Here, we used the in-order traversal of depth-first search algorithm on the tree-structured topology. The output matrix is composed of the hidden states of all non-leaf nodes, defined as $\mathbf{H}=[\mathbf{h}_1,\mathbf{h}_2,...,\mathbf{h}_T] \in \Re^{T\times d_h}$, where T and d_h respectively denote the number and dimensionality of the hidden states. The obtained hidden matrix is then fed to the aggregation layer.

2.2 Dynamic Routing

To compose all word vectors to generate sentence vectors, the tree-structured LSTM model uses the hidden states of all non-leaf nodes to obtain the weights for all nodes through the dynamic routing algorithm.

Taking the hidden states of all non-leaf nodes as the input vectors, the goal of dynamic routing is to encode the sentiment information of those vectors into a fixed-length sentence vector,

$$\mathbf{s}_{cap} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_J] \tag{5}$$

Inspired by the definition of capsule networks, we implement two layers of capsules (i.e., $\mathbf{H}=[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_I]$ denotes the input capsules and $\mathbf{s}=[\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_J]$ denotes the output capsules) to perform dynamic routing. The output capsule $\mathbf{s}_j$ is produced from a non-linear "squashing" function to ensure $|\mathbf{s}_j| \in (0,1)$ as a probability,

$$\mathbf{s}_j = \frac{||\mathbf{v}_j||^2}{1+||\mathbf{v}_j||^2}\frac{\mathbf{v}_j}{||\mathbf{v}_j||} \tag{6}$$

where $\mathbf{s}_j$ is the vector output of capsule j, $\mathbf{v}_j$ is the total input, which is a weighted sum over all "prediction vectors" $\hat{\mathbf{h}}_{j|t}$ from the capsules in the layer below,

$$\mathbf{v}_j = \sum_t c_{tj}\hat{\mathbf{h}}_{j|t} \tag{7}$$

where coupling coefficients c_{tj} are the probability distributions of capsule j which are computed using a softmax function so that all capsules in the layer above sum to 1 so that the sentiment information $\hat{\mathbf{h}}_{j|t}$ is obtained by multiplying the input vector $\mathbf{h}_t$ by a weighted matrix W_{tj}, defined as,

$$\hat{\mathbf{h}}_{j|t} = W_{tj}\mathbf{h}_t \tag{8}$$

Here, the coupling coefficients c_{tj} are determined by the iterative dynamic routing process,

$$c_{tj} = \frac{\exp(b_{tj})}{\sum_k \exp(b_{tk})} \tag{9}$$

$$b_{tj} \coloneqq b_{tj} + \hat{\mathbf{h}}_{j|t}\cdot\mathbf{s}_j \tag{10}$$

where b_{tj} is the log probabilities, initialized with 0. The detailed iterative process of learning the weights between capsules in two layers for each non-leaf node is shown in Fig. 2.

In Eq. (7), the capsules in the above layer try to learn contribution weights c_{tj} (i.e., coupling coefficients) for the capsules in the below layer. The updated information in b_{tj} comes from the scalar product $\hat{\mathbf{h}}_{j|t}\cdot\mathbf{s}_j$. The coupling coefficients c_{ij} are iteratively refined by measuring the agreement between the current output $\mathbf{s}_j$ of output capsule j in the above layer and the prediction $\hat{\mathbf{h}}_{j|t}$ made by input capsule i. If the margin between the two vectors and $\mathbf{s}_j$ is very large, the scalar product of

SST (Classification)	Binary	Fine-grained
CNN	87.2	48.0
GRU	87.2	48.2
LSTM	84.9	46.4
Bi-GRU	87.4	48.5
Bi-LSTM	87.5	49.1
2-Layer Bi-GRU	87.1	48.7
2-Layer Bi-LSTM	87.2	48.5
Tree-LSTM	87.5	49.7
Attention GRU	87.8	49.5
Attention LSTM	87.6	49.2
Attention Tree-LSTM	88.2	49.8
Capsule Tree-LSTM	**90.2**	**51.6**

Table 1: Results of different methods on SST.

EmoBank (Regression)	Valence		Arousal	
	MAE	*r*	MAE	*r*
CNN	0.581	0.521	0.560	0.519
GRU	0.523	0.589	0.527	0.532
LSTM	0.518	0.592	0.528	0.534
Bi-GRU	0.514	0.591	0.497	0.543
Bi-LSTM	0.506	0.610	0.498	0.578
2-Layer Bi-GRU	0.505	0.612	0.485	0.573
2-Layer Bi-LSTM	0.498	0.615	0.475	0.588
Tree-LSTM	0.483	0.625	0.468	0.602
Attention GRU	0.492	0.622	0.477	0.585
Attention LSTM	0.495	0.620	0.472	0.589
Attention Tree-LSTM	0.475	0.629	0.465	0.596
Capsule Tree-LSTM	**0.462**	**0.639**	**0.454**	**0.622**

Table 2. Results of different methods on EmoBank.

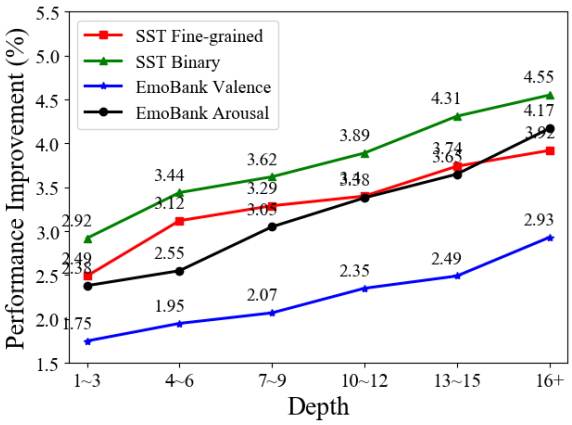

Figure 3: Effect of dynamic routing algorithm.

those vectors will be large, which will also result in an update on the coupling coefficient c_{tj}.

3 Experimental Results

Datasets. This experiment used two datasets for evaluation. i) The Stanford Sentiment Treebank (SST) (Socher et al., 2013) is used for sentiment classification. It contains 6920/872/1821 sentences for the train/dev/test sets with binary labels (positive/negative) and 8544/1101/2210 sentences with fine-grained labels (very negative/negative/ neutral/positive/very positive). ii) EmoBank (Buechel and Hahn, 2017; Buechel and Hahn, 2016) is used for sentiment regression to predict valence-arousal (VA) values (Wang et al., 2016b; Yu et al., 2016). It contains 10,000 sentences with real-valued VA ratings in the range of (1, 9), where the valence refers to the degree of positive and negative sentiment and the arousal refers to the degree of calm and excitement. The provided ratings have Reader and Writer perspectives, and the Reader was adopted as the ground-truth ratings due to its superiority reported in (Buechel and Hahn, 2017). We performed 5-fold cross-validation (6:2:2) on the EmoBank dataset.

Evaluation Metrics. For SST, the evaluation metric is accuracy for both binary and fine-grained classification. For EmoBank, we used the Pearson correlation coefficient (*r*) and mean absolute error (MAE). A higher *r* or a lower MAE value indicates better prediction performance.

Implementation Details. Several deep neural networks were implemented for comparison, including CNN, GRU, LSTM, and tree-LSTM. For the sequential models (GRU and LSTM), we additionally implemented an enhanced version using a bi-directional strategy and 2-layer stacked architecture. To investigate the performance of self-attention, we also implement a self-attention layer by taking as input the hidden states of all non-leaf nodes, to form an attention Tree-LSTM model (Kokkinos and Potamianos, 2017). For word vectors, we used GloVe pre-trained on the 840B Common Crawl corpus (Pennington et al., 2014). The respective dimensionality values of the word vectors and hidden states were 300 and 120. For classification and regression tasks, *softmax* and *linear decoder* (Wang et al., 2016a) activation function are respectively applied as the output layer.

Comparative Results. Tables 1 and 2 respectively show the comparative results of different methods for SST and EmoBank. Both the enhanced bi-directional and 2-layer GRU/LSTM outperformed the standard GRU, LSTM, CNN, and the Tree-LSTM with structural information achieved better performance than all of them for both classification and regression tasks. Once the dynamic routing algorithm was introduced, the proposed Capsule Tree-LSTM further improved the performance of Tree-LSTM

(with attention).

Figure 3 shows the detailed analysis of the effect of dynamic routing. The test sentences were first divided into several groups according to their depths in the parse trees (e.g., the depth of the example sentence in Fig. 1 is three). The performance improvement of Capsule Tree-LSTM over Tree-LSTM was then calculated for each group. The results show that the performance improvements increased with the increase of the depth. The reason is that the Tree-LSTM may suffer from a more serious bias problem for sentences with a deeper tree structure because the useful nodes in the deeper levels tend to be ignored. Conversely, the Capsule Tree-LSTM can assign a higher weight to the nodes that contribute more to the prediction even though they lie in the leaf node at the bottom of the tree.

4 Conclusion

This study presents a capsule tree-LSTM model for sentiment classification and regression. The proposed method uses dynamic routing algorithm to automatically learn the weights of each node to compose sentence representations. Experimental results show that the proposed method yielded better results than convolutional (CNN), sequential (LSTM and GRU), structural (tree-LSTM) and self-attention neural networks. Future work will conduct more detailed analysis to continue enhancing the proposed method.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61966038, 61702443 and 61762091, and in part by the Ministry of Science and Technology, Taiwan, ROC, under Grant No. MOST107-2628-E-155-002-MY3 and MOST107-2218-E-006-008. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Sven Buechel and Udo Hahn. 2016. Readers vs. writers vs. texts : Coping with different perspectives of text understanding in emotion annotation. In *Proceedings of the 11th Linguistic Annotation Workshop in EACL 2017*, pages 1–12.

Sven Buechel and Udo Hahn. 2017. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2017)*, pages 578–585.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Alex Graves. 2012. *Supervised sequence labelling*. Springer Berlin Heidelberg.

Minlie Huang, Qiao Qian, and Xiaoyan Zhu. 2017. Encoding syntactic knowledge in neural networks for sentiment classification. *ACM Transactions on Information Systems*, 35(3):1–27.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, pages 720–728.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-2015)*, pages 1681–1691.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-2014)*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of International Conference of Empirical Methods on Natural Language Processing (EMNLP-2014)*, pages 121–129.

Filippos Kokkinos and Alexandros Potamianos. 2017. Structural attention neural networks for improved sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2017)*, pages 586–591.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems (NIPS-2013)*.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013b. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR-2013)*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, pages 1532–1543.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS-2017)*, pages 3859–3869.

Richard Socher, Alex Perelygin, and Jy Wu. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP-2013)*, pages 1631–1642.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-2014)*, pages 1556–1566.

Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. 2016. Sentiment embeddings with applications to sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(2):496–509.

Jin Wang, Liang-Chih Yu, K.Robert Lai, and Xuejie Zhang. 2016a. Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-2016)*, pages 225–230.

Jin Wang, Liang Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016b. Community-based weighted graph model for valence-arousal prediction of affective words. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 24(11):1957–1968.

Xin Wang, Yuanchao Liu, Chengjie Sun, Baoxun Wang, and Xiaolong Wang. 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-2015)*, pages 1343–1353.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT-16)*, pages 1480–1489.

Liang-Chih Yu, Lung-Hao Lee, Shuai Hao, Jin Wang, Yunchao He, Jun Hu, K Robert Lai, and Xuejie Zhang. 2016. Building Chinese affective resources in valence-arousal dimensions. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT-2016)*, pages 540–545.

Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xuejie Zhang. 2018a. Refining word embeddings for sentiment analysis. In *Proceedings ofthe 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP-2017)*, pages 534–539.

Liang-Chih Yu, Jin Wang, K.Robert Lai, and Xuejie Zhang. 2018b. Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 26(3):671–681.