2023 IEEE 3rd International Conference on Computer Communication and Artificial Intelligence (CCAI) | 979-8-3503-3526-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCAI57533.2023.10201311

Hyperbolic Graph Convolutional Networks for Aspect-Based Sentiment Analysis

Xueda Li
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
xueda@mail.dlut.edu.cn

Changrong Min
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
11909060@mail.dlut.edu.cn

Hao Zhang
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
zh373911345@mail.dlut.edu.cn

Liang Yang
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
liang@dlut.edu.cn

Dongyu Zhang
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
zhangdongyu@dlut.edu.cn

Hongfei Lin*
School of Computer Science and Technology
Dalian University of Technology
Dalian, China
hflin@dlut.edu.cn

***Abstract*—Aspect-based sentiment analysis is a fine-grained sentiment analysis task that aims to predict the sentiment polarity of a specific aspect. Recent work adopts graph convolutional networks over dependency trees to capture the syntactic connections of aspects and opinion words while introducing the BiAffine to jointly refine syntax structures and semantic correlations. However, in the Euclidean space, the neural network models can't well capture the syntactic connections of aspects and opinion words due to the inaccurate dependency trees representation, and the original structures and correlations are affected due to the BiAffine exchange method. Fortunately, dependency trees can be represented well since hyperbolic space can be viewed as continuous simulations of trees, so we propose a hyperbolic graph convolutional networks (HyperGCN) model to handle these challenges. We employ hyperbolic graph convolution with the dependency tree to model syntactic connections between aspects and opinion words, additionally, we also capture the semantic correlations with a hyperbolic graph convolutional network incorporating self-attention mechanism. Particularly, to exchange the relevant features without original syntax structures and semantic correlations being affected, we leverage an attention mechanism with residual structure to exchange relevant features of syntactic and semantic information. The experimental results on three datasets verify the effectiveness of our model.**

Keywords*—*HyperGCN, ABSA, attention mechanism, residual structure

I. INTRODUCTION

Aspect-based sentiment analysis (ABSA) is an entity-oriented fine-grained sentiment analysis task that aims to determine the sentiment polarity of a given aspect in a sentence. In recent years, plenty of work begin to introduce dependency trees and apply neural networks to model the relations between sentences and aspects. Most of these methods apply graph convolutional network and graph attention network over dependency trees. Among them, Li et al. [1] propose a dual graph convolutional network and employ a BiAffine module for the feature interchange between syntactic structures and semantic correlations to obtain a better representation respectively. But the dependency trees are tree-like structures, and in Euclidean space, the traditional neural network model can't effectively represent the complex tree-like hierarchies due to the limitation of the number of parameters [2]. At the same time, the BiAffine module has the problem that the original syntactic structures and semantic correlations will be affected during feature exchange, for example, the semantic correlations information is affected on account of the introduction of too many irrelevant syntactic features. Fortunately, hyperbolic geometry provides a perfect solution. The hyperbolic space can be viewed as a continuously simulated tree, so we can easily leverage hyperbolic geometry to embed the trees with low distortion [3]. Chami et al. [4] use hyperbolic graph convolutional networks for link prediction and node classification tasks and achieves good performance on multiple datasets.

Inspired by their work, we introduce the hyperbolic graph convolutional network into the ABSA task and propose a hyperbolic graph convolutional network model based on the dual graph convolutional network, and the experiment proves the effectiveness of our proposed model, Our contributions are summarized as follows:

- We effectively represent the hierarchies of dependency trees with hyperbolic geometry, thus modeling the syntactic connections between opinion words and aspects better.

- We design the residual attention mechanism for ensuring that the original syntactic structures and semantic correlations are not affected when exchanging features.

- We conduct extensive experiments on multiple datasets to verify the model's effectiveness.

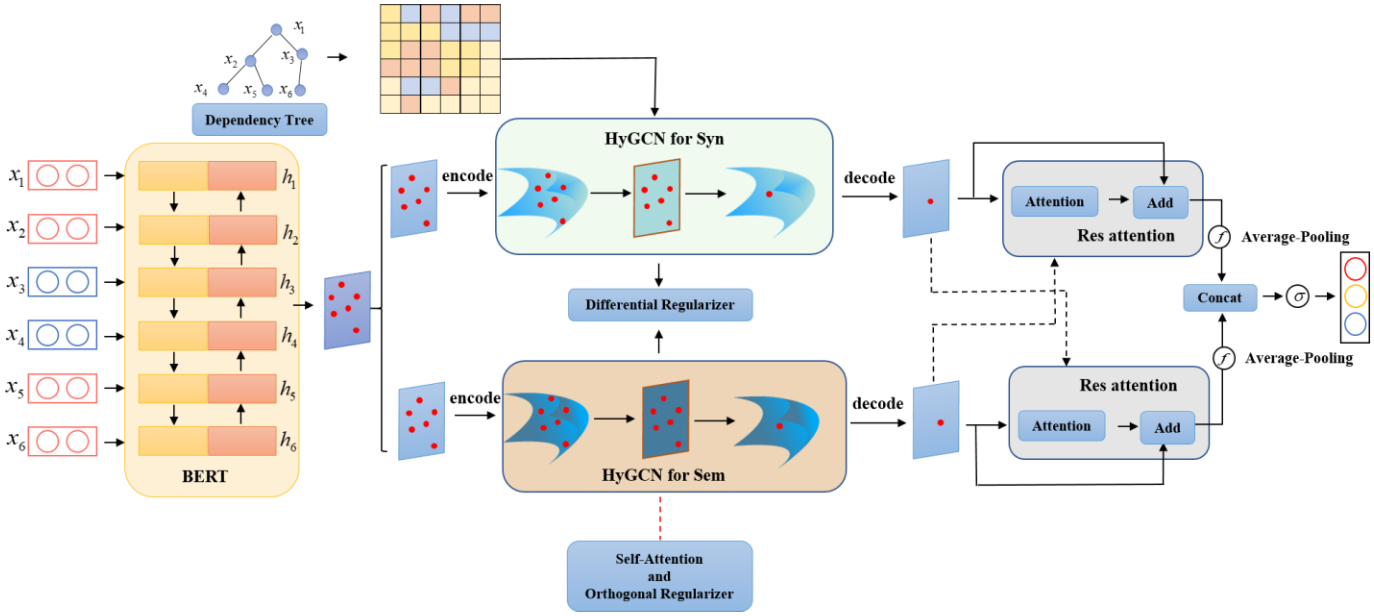

Fig. 1. The framework of HyperGCN. HyGCN for Syn is the hyperbolic graph convolutional networks for extracting syntactic structures and HyGCN for Sem is the hyperbolic graph convolutional networks for extracting semantic correlations. Res attention is the module of attention mechanism with residual structure.

In recent years, most of the recent methods have leverage neural networks to model the relations of the given aspects and the contexts. Among these methods, there are some outstanding works that have obtained relatively good results. For instance, Wang et al. [5] propose an Attention-based Long Short-Term Memory Network for aspect-level sentiment classification. However, the state-of-the-art model should be the graph neural networks over dependency trees. Chen et al. [6] propose a discrete latent opinion tree model as an alternative structure to explicit dependency trees.

B. Hyperbolic Neural Networks

However, these methods over dependency trees only represent the dependency trees in the Euclidean space, ignoring the problem that the Euclidean space cannot represent the tree structure of the dependency trees well. Fortunately, hyperbolic space is a space similar to tree structure, and it is good at representing hierarchical structure, so it can represent the dependency trees well. Thus, hyperbolic geometry becomes a good alternative. Meanwhile, a variety of hyperbolic neural networks have been proposed and widely used recently. Sonthalia and Gilbert. [7] explore a new method for learning hyperbolic representations, this method can preferentially learns the tree structure of the data. Chen et al. [8] design a hyperbolic capsule networks for multi-label classification, and applies the hyperbolic neural networks to text classification.

III. Preliminaries

We need to introduce some necessary knowledge of hyperbolic geometry in this section.

II. Related Work

A. Aspect-based sentiment analysis

Hyperbolic manifold. L^n is a n dimension and continuous negative curvature $-c(c>0)$ hyperbolic manifold embedded in an $n+1$ dimension Minkowski space.

$$L^{n,c}=\{x=(x_0,x_1,\cdots,x_n)\in R^{n+1}:\\ <x,x>_L=-\frac{1}{c},x_0>0\} \tag{1}$$

where $<,>\in R^{n+1}\times R^{n+1}$ is the Minkowski inner product, it is defined as:

$$<x,y>_L=-x_0y_0+\sum_{k=1}^{n}x_ky_k \tag{2}$$

Tangent space. The centered on point x tangent space can be defined as:

$$\tau_x L^{n,c}=\{v=(v_0,v_1,\cdots v_n)\in R^{n+1}:<v,x>_L=0\} \tag{3}$$

Geodesic. Geodesic is generally used to represent distances in manifold space, it can be expressed as:

$$d_L^c(x,y)=\frac{ar\cosh(-<x,y>_L c)}{\sqrt{c}} \tag{4}$$

Exponential and logarithmic maps. The mapping between tangent space and hyperbolic space generally adopts exponential map $\exp_x^c(v)$ and logarithmic map $\log_x^c(y)$. They can be expressed as:

$$\exp_x^c(v) = \cosh(\sqrt{c}\,\| v \|_L)x + \frac{v \sinh(\sqrt{c}\,\| v \|_L)}{\| v \|_L \sqrt{c}} \quad (5)$$

$$\log_x^c(y) = d_L^c(x,y)\frac{y + c < x, y >_L x}{\| y + c < x, y >_L x \|_L} \quad (6)$$

IV. PROPOSED MODEL

In this section, we will introduce our HyperGCN model. Figure 1 shows the overall architecture of our model.

A. Hyperbolic Graph Convolutional Networks

BERT encoding. For the BERT encoding, given an aspect-sentence pair (a,s) where the sentence $s = \{w_1, w_2, \cdots, w_n\}$ contains $a = \{a_1, a_2, \cdots, a_m\}$ .We input "$[CLS]sentence[SEP]aspect[SEP]$" into the BERT model to obtain the hidden state representation $H = \{h_1, h_2, \cdots, h_n\}$.

Manifold encoding. Let $x^{0,E}$ to denote the representation encoded by BERT. Then we need to use exponential mapping to embed $x^{0,E}$ from Euclidean space into hyperbolic space , we set the origin as:

$$o = (\frac{1}{\sqrt{c}}, 0, \cdots, 0) \in L^{n,c}, and < (0, x^{0,E}), o >= 0 \quad (7)$$

Then the embedding after mapped $x^{0,L}$ can be obtained:

$$x^{0,L} = \exp_o^c((0, x^{0,E})) \quad (8)$$

Hyperboloid linear transform. For the transformation, we first use the logarithmic mapping method to project the hyperbolic point x^L to the tangent space $\tau_o L^{j,c}$. Then we conduct hyperbolic matrix multiplication and hyperbolic paranoid addition, the two formulas are respectively defined as:

$$W \otimes^c x^L = \exp_o^c(W \log_o^c(x^H)) \quad (9)$$

$$x^H \oplus^c b = \exp_{x^L}^c(T_{0 \to x^L}^c(b)) \quad (10)$$

where W is weight matrix and b is bias.

Hyperboloid neighborhood aggregation. In hyperbolic space, for the hyperbolic embeddings (x_i^L, x_j^L) that have been obtained, we also first need to use logarithmic mapping to map x_i^L and x_j^L into the tangent space of the origin, then we adopts the hyperbolic aggregation to gain the nodes' representations.

$$AGG^c(x^L)_i = \exp_{x_i^L}^L(\sum_{j=1}^{n} A_{ij} \log_{x_i^L}^c(x_j^L)) \quad (11)$$

where $A \in R^{n \times n}$ is the adjacency matrix input

Hyperboloid non-linear activation. Let c_{l-1} denotes the curvature of the layer $l-1$ and c_l denotes the curvature of the layer l . We adopt a nonlinear activation function in the tangent space $\tau_o L^{n,c_{l-1}}$, and then map back to the hyperbolic manifold L^{n,c_l} , the formula is as follows:

$$\sigma^{\otimes^{c_{l-1},c_l}}(x^L) = \exp_o^{c_l}(\sigma(\log_o^{c_{l-1}}(x^L))) \quad (12)$$

Manifold decoding. We embed the output into the tangent space of the origin by the logarithmic map for Euclidean space operations. The formula is defined as:

$$x^{0,\tau} = \log_o^{c_l}(x^{l,L}) \quad (13)$$

Hyperboloid graph convolutional networks. We have introduced all the components of the hyperbolic graph convolutional network in the previous content, so the structure of graph convolution operation is as follows:

$$\begin{aligned} h_i^{l,L} &= (W^l \otimes^{c_{l-1}} x_i^{l-1,L}) \oplus^{c_{l-1}} b^l \\ y_i^{l.L} &= AGG^{c_{l-1}}(h^{l,L})_i \\ x_i^{l,L} &= \sigma^{\otimes^{c_{l-1},c_l}}(y_i^{l.L}) \end{aligned} \quad (14)$$

Finally, we embed the last layer output of hyperbolic graph convolution into the tangent space of the origin by manifold decoding, and then we can do operations in Euclidean space.

B. Dual Hyperbolic Graph Convolutional Networks

We use two hyperbolic graph convolutional networks to capture semantic association and syntactic structure.

For semantic association extraction, we use the self-attention matrix as the adjacency matrix for input. Let $A^{se} \in R^{n \times n}$ denote the self-attention matrix:

$$A^{se} = soft\max(\frac{QW_1 \times (KW_2)^T}{\sqrt{n}}) \quad (15)$$

where Q and K are both BERT encoded representations, W_1 and W_2 are learnable weight matrices, n is the input dimension. Then we employ orthogonal regularizer [1] for orthogonality among the attention score vectors of all words and make it as a part of the loss.

$$L_{ort} = \| A^{se} A^{seT} - I \|_F \quad (16)$$

where I is an identity matrix.

Finally, we will get the semantic correlations hidden representations $H^{se} = (h_1^{se}, h_2^{se}, \cdots h_n^{se})$, and $h_i^{se} \in R^n$ denotes hidden representation of node i. We will also do the same operations for aspect nodes to get aspect semantic correlations hidden representation $H_a^{se} = (h_{a_1}^{se}, h_{a_2}^{se}, \cdots h_{a_n}^{se})$. For syntactic structures, we adopt LAL-Parser [9] as the dependency parsing model and input the obtained dependency probability matrix $A^{sy} \in R^n \times R^n$ to the hyperbolic graph convolutional network. The hyperbolic graph convolutional networks can perfectly represent dependency trees with low distortion because of the tree-like structure of hyperbolic spaces. For learning different semantic correlations and syntactic structures information with hyperbolic graph convolutional network, we employ differential regularizer [1] and make it as a part of the loss.

$$L_{dif} = \frac{1}{\| A^{se} - A^{sy} \|_F} \tag{17}$$

We also get syntactic structures hidden representation $H^{sy} = (h_1^{sy}, h_2^{sy}, \cdots h_n^{sy})$ and aspect syntactic hidden representation $H_a^{sy} = (h_{a_1}^{sy}, h_{a_2}^{sy}, \cdots h_{a_n}^{sy})$.

C. Attention Mechanism with Residual Structure

We leverage attention mechanism with residual structure to exchange the relevant features without original syntax structures and semantic correlations being affected. We process semantic correlations hidden representation and syntactic structures hidden representation as follows:

$$\begin{aligned} H^{se'} &= soft\max(\frac{H^{se}W_3(H^{sy}W_4)^T}{\sqrt{n}})H^{sy}W_5 \\ H^{se'} &= H^{se'} + H^{se} \end{aligned} \tag{18}$$

$$\begin{aligned} H^{sy'} &= soft\max(\frac{H^{sy}W_6(H^{se}W_7)^T}{\sqrt{n}})H^{se}W_8 \\ H^{sy'} &= H^{sy'} + H^{sy} \end{aligned} \tag{19}$$

where W_3, W_4, W_5, W_6, W_7 and W_8 are trainable weight matrix.

Finally, we apply average pooling on H_a^{se} and H_a^{sy}:

$$\begin{aligned} h_a^{se} &= AveragePooling(H_a^{se}) \\ h_a^{sy} &= AveragePooling(H_a^{sy}) \end{aligned} \tag{20}$$

For BERT encoding, we concat h_a^{se}, h_a^{sy} and $[CLS]$ after pooling:

$$f = Concat(h_a^{se}, h_a^{sy}, h_{CLS}) \tag{21}$$

We compute the probability contribution of sentiment applying a linear layer and a softmax function:

$$p(a) = soft\max(W_p f + b_p) \tag{22}$$

where W_p and b are trainable weight matrix and bias.

D. Loss Function

We norm all trainable model parameters as part of the loss function:

$$L_{par} = \sqrt{\theta_1^2 + \theta_2^2 + \cdots + \theta_t^2} \tag{23}$$

where θ_i is trainable model parameter and t is the number of trainable model parameters.

We also apply the cross entropy loss function as part of the loss function:

$$L_{cross} = -\sum_{(s,a)\in D} \sum_{c \in C} \log p(a) \tag{24}$$

where D is a collection of aspect-sentence pairs, C is the collection of distinct sentiment classes.

Finally, we combine all the loss function parts as our final loss function to train the model better:

$$L_{total} = L_{cross} + \lambda_1 L_{ort} + \lambda_2 L_{dif} + \lambda_3 L_{par} \tag{25}$$

where λ_1, λ_2 and λ_3 are regularization coefficient.

V. EXPERIMENTS

A. Datasets

We do experiments on some public datasets including Restaurant14, Laptop14 and Twitter. Both Restaurant and Laptop are from SemEval 2014 [14] that are comments on Laptop and Restaurant. Following [10], we delete instances using the "conflict" label. The Twitter dataset [15] is also a very important dataset for us, which is mainly composed of some tweets.

B. Implementation and Parameter Settings

Our initial word embeddings, dependency parser, and input to BERT all follow [1]. Furthermore, following [16], we set up a self-loop for each node in the hyperbolic graph convolutional network.

C. Baselines

To verify the performance of our proposed model (HyperGCN), we compare HyperGCN with multiple baselines, including DGEDT+BERT [11], DualGCN+BERT [1], BERT[12], T-GCN+BERT [13], dotGCN+BERT [6].

TABLE I. THE COMPARISON PERFORMANCE BETWEEN HYPERGCN AND BASELINES

Models	Restaurant14		Laptop 14		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
BERT [12]	86.15	80.29	81.01	76.69	75.18	74.01
DGEDT+BERT [11]	86.30	80.00	79.80	75.60	77.90	75.40
DualGCN+BERT [1]	87.13	81.16	**81.80**	78.10	77.40	76.02
T-GCN+BERT [13]	86.16	79.95	80.88	77.03	76.45	75.25
dotGCN+BERT [6]	86.16	80.49	81.03	78.10	78.11	77.00
Our HyperGCN+BERT	**87.49**	**82.19**	80.85	**78.18**	**78.14**	**77.16**

TABLE II. ABLATION EXPERIMENT RESULTS

Models	Restaurant14		Laptop14		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
w/o attention	86.51	80.96	80.06	76.68	77.40	76.11
w/o hyperbolic	85.70	79.73	78.96	74.36	76.66	75.21
Our HyperGCN +BERT	**87.49**	**82.19**	**80.85**	**78.18**	**78.14**	**77.16**

D. Results

Table I shows the comparison results between our model(HyperGCN) and all Baselines. We evaluate all models with Accuracy and Macro-F1. In the experiment, we use BERT as the encoding method. The analyses of the results are as follows:

Compared with the BERT model[12], we obtain an improvements of 3.15, 1.9 and 1.49 Macro-F1 score on Twitter, Restaurant and Laptop datasets respectively. We also have an improvements on the accuracy score, particularly, 2.96 accuracy score is improved on Twitter dataset. At the same time, in the Restaurant and Laptop datasets, the dimension of our final representation (below 80 dimensions) is much lower than the 768 dimensions of other baselines, which verifies the high efficiency of hyperbolic neural network representation in low-dimensional space. In addition, our model also performs better than the baselines (DGEDT+BERT [11] , DualGCN+BERT [1]) that employ BiAffine for feature interchange, it shows that the attention mechanism with residual structure we adopt is more effective than BiAffine.

E. Ablation Study

We conduct extensive ablation experiments for verifying the effectiveness of each module, and the experiment results are shown in Table II. We conduct ablation experiments on BERT encoding, we first ablate the attention mechanism, which causes the syntactic structures and semantic associations without exchanging features, and we can see a significant drop in BERT encoding experiment. We also use Euclidean GCN instead of hyperbolic GCN module, which will cause the dependency tree to not be well represented. This directly leads to a large drop in performance on all datasets, such as a drop of 2.85 accuracy score and 3.59 Macro-F1 score on Laptop

VI. CONCLUSION

In this paper, we propose a HyperGCN model which can represent the dependency trees with low distortion in the hyperbolic space, overcoming the problem that the traditional neural network can't represent the complex dependency trees well due to the parameter number limitation in the Euclidean space. Syntactic structures and semantic correlations are well extracted by two hyperbolic graph convolutional networks. Furthermore, we use a residual-structured Attention mechanism to exchange syntactic structures and semantic correlations relevant features without destroying the original information. Extensive experimental results show that our model is effective and outperforms graph convolutional networks in Euclidean space.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (No.62076046).

REFERENCES

[1] Li, R., Chen, H., Feng, F., Ma, Z., Wang, X., Hovy, E.: Dual graph convolutional networks for aspect-based sentiment analysis. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume1: Long Papers), pp. 6319–6329 (2021)

[2] Zhu, Y., Zhou, D., Xiao, J., Jiang, X., Chen, X., Liu, Q.: Hypertext: Endowing fasttext with hyperbolic geometry. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 1166–1171 (2020)

[3] Krioukov D, Papadopoulos F, Kitsak M, et al. Hyperbolic geometry of complex networks[J]. Physical Review E, 2010, 82(3): 036106.

[4] Chami I, Ying Z, Ré C, et al. Hyperbolic graph convolutional neural networks[J]. Advances in neural information processing systems, 2019, 32.

[5] Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based lstm for aspect-level sentiment classification. In: Proceedings of the 2016

Conference on Empirical Methods in Natural Language Processing, pp. 606–615 (2016)

[6] Chen, C., Teng, Z., Wang, Z., Zhang, Y.: Discrete opinion tree induction for aspect-based sentiment analysis. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.2051–2064 (2022)

[7] Sonthalia, R., Gilbert, A.: Tree! i am no tree! i am a low dimensional hyperbolic embedding. Advances in Neural Information Processing Systems 33, 845–856(2020)

[8] Chen B, Huang X, Xiao L, et al. Hyperbolic capsule networks for multi-label classification[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 3115-3124.

[9] Mrini, K., Dernoncourt, F., Bui, T., Chang, W., Nakashole, N.: Rethinking self-attention: An interpretable self-attentive encoder-decoder parser (2019)

[10] Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 452–461 (2017)

[11] Tang, H., Ji, D., Li, C., Zhou, Q.: Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6578–6588 (2020)

[12] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)

[13] Tian, Y., Chen, G., Song, Y.: Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2910–2922 (2021)

[14] Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos,J., Manandhar, S.: Semeval-2014 task 4: Aspect based sentiment analysis

[15] Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 2: Short Papers), pp. 49–54 (2014)

[16] Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1506–1515 (2017)