Introducing Syntactic Structures into Target Opinion Word Extraction with Deep Learning

Amir Pouran Ben Veyseh[1]*, **Nasim Nouri***, **Franck Dernoncourt**[2], **Dejing Dou**[1] and **Thien Huu Nguyen**[1,3]
[1] Department of Computer and Information Science, University of Oregon, Eugene, OR 97403, USA
[2] Adobe Research, San Jose, CA, USA
[3] VinAI Research, Vietnam
{apouranb,dou,thien}@cs.uoregon.edu,
nasim.nourii@gmail.com, dernonco@adobe.com

Abstract

Targeted opinion word extraction (TOWE) is a sub-task of aspect based sentiment analysis (ABSA) which aims to find the opinion words for a given aspect-term in a sentence. Despite their success for TOWE, the current deep learning models fail to exploit the syntactic information of the sentences that have been proved to be useful for TOWE in the prior research. In this work, we propose to incorporate the syntactic structures of the sentences into the deep learning models for TOWE, leveraging the syntax-based opinion possibility scores and the syntactic connections between the words. We also introduce a novel regularization technique to improve the performance of the deep learning models based on the representation distinctions between the words in TOWE. The proposed model is extensively analyzed and achieves the state-of-the-art performance on four benchmark datasets.

1 Introduction

Targeted Opinion Word Extraction (TOWE) is an important task in aspect based sentiment analysis (ABSA) of sentiment analysis (SA). Given a target word (also called aspect term) in the input sentence, the goal of TOWE is to identify the words in the sentence (called the target-oriented opinion words) that help to express the attitude of the author toward the aspect represented by the target word. For instance, as a running example, in the sentence "*All warranties honored by XYZ (what I thought was a reputable company) are disappointing.*", "*disappointing*" is the opinion word for the target word "*warranties*" while the opinion words for the target word "*company*" would involve "*reputable*". Among others, TOWE finds its applications in target-oriented sentiment analysis (Tang et al., 2016; Xue and Li, 2018; Veyseh et al., 2020) and opinion summarization (Wu et al., 2020).

*Equal contribution.

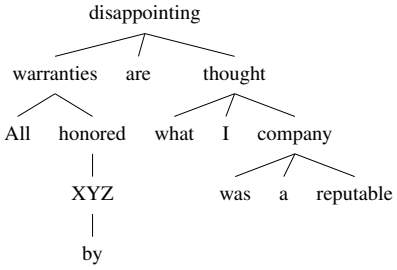

Figure 1: The dependency tree of the example sentence.

The early approach for TOWE has involved the rule-based and lexicon-based methods (Hu and Liu, 2004; Zhuang et al., 2006) while the recent work has focused on deep learning models for this problem (Fan et al., 2019; Wu et al., 2020). One of the insights from the rule-based methods is that the syntactic structures (i.e., the parsing trees) of the sentences can provide useful information to improve the performance for TOWE (Zhuang et al., 2006). However, these syntactic structures have not been exploited in the current deep learning models for TOWE (Fan et al., 2019; Wu et al., 2020). Consequently, in this work, we seek to fill in this gap by extracting useful knowledge from the syntactic structures to help the deep learning models learn better representations for TOWE. In particular, based on the dependency parsing trees, we envision two major syntactic information that can be complementarily beneficial for the deep learning models for TOWE, i.e., the syntax-based opinion possibility scores and syntactic word connections for representation learning. First, for the syntax-based possibility scores, our intuition is that the closer words to the target word in the dependency tree of the input sentence tend to have better chance for being the opinion words for the target in TOWE. For instance, in our running example, the opinion word "*disappointing*" is sequentially far from its target word "*warranties*". However, in the de-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 8947–8956,
November 16–20, 2020.

pendency tree shown in Figure 1, "*disappointing*" is directly connected to "*warranties*", promoting the distance between "*disappointing*" and "*warranties*" (i.e., the length of the connecting path) in the dependency tree as an useful feature for TOWE. Consequently, in this work, we propose to use the distances between the words and the target word in the dependency trees to obtain a score to represent how likely a word is an opinion word for TOWE (called syntax-based possibility scores). These possibility scores would then be introduced into the deep learning models to improve the representation learning for TOWE.

In order to achieve such possibility score incorporation, we propose to employ the representation vectors for the words in the deep learning models to compute a model-based possibility score for each word in the sentence. The model-based possibility scores also aim to quantify the likelihood of being an opinion word for each word in the sentence; however, they are based on the internal representation learning mechanism of the deep learning models for TOWE. To this end, we propose to inject the information from the syntax-based possibility scores into the models for TOWE by enforcing the similarity/consistency between the syntax-based and model-based possibility scores for the words in the sentence. The rationale is to leverage the possibility score consistency to guide the representation learning process of the deep learning models (using the extracted syntactic information) to generate more effective representations for TOWE. In this work, we employ the Ordered-Neuron Long Short-Term Memory Networks (ON-LSTM) (Shen et al., 2019) to obtain the model-based possibility scores for the words in the sentences for TOWE. ON-LSTM introduces two additional gates into the original Long Short-Term Memory Network (LSTM) cells that facilitate the computation of the model-based possibility scores via the numbers of active neurons in the hidden vectors for each word.

For the second type of syntactic information in this work, the main motivation is to further improve the representation vector computation for each word by leveraging the dependency connections between the words to infer the effective context words for each word in the sentence. In particular, motivated by our running example, we argue that the effective context words for the representation vector of a current word in TOWE involve the neighboring words of the current word and the target word in the dependency tree. For instance, consider the running example with "*warranties*" as the target word and "*reputable*" as the word we need to compute the representation vector. On the one hand, it is important to include the information of the neighboring words of "*reputable*" (i.e., "*company*") in the representation so the models can know the context for the current word (e.g., which object "*reputable*" is modifying). On the other hand, the information about the target word (i.e., "*warranties*" and possibly its neighboring words) should also be encoded in the representation vector for "*reputable*" so the models can be aware of the context of the target word and make appropriate comparison in the representation to decide the label (i.e., non-opinion word) for "*reputable*" in this case. Note that this syntactic connection mechanism allows the models to de-emphasize the context information of "*I*" in the representation for "*reputable*" to improve the representation quality. Consequently, in this work, we propose to formulate these intuitions into an importance score matrix whose cells quantify the contextual importance that a word would contribute to the representation vector of another word, given a target word for TOWE. These importance scores will be conditioned on the distances between the target word and the other words in the dependency tree. Afterward, the score matrix will be consumed by a Graph Convolutional Neural Network (GCN) model (Kipf and Welling, 2017) to produce the final representation vectors for opinion word prediction.

Finally, in order to further improve the induced representation vectors for TOWE, we introduce a novel inductive bias that seeks to explicitly distinguish the representation vectors of the target-oriented opinion words and those for the other words in the sentence. We conduct extensive experiments to demonstrate the benefits of the proposed model, leading to the state-of-the-art performance for TOWE in several benchmark datasets.

2 Related Work

Comparing to the related tasks, TOWE has been relatively less explored in the literature. In particular, the most related task of TOWE is opinion word extraction (OWE) that aims to locate the terms used to express attitude in the sentences (Htay and Lynn, 2013; Shamshurin, 2012). A key difference between OWE and TOWE is that OWE does not require the opinion words to tie to any target words

in the sentence while the opinion words in TOWE should be explicitly paired with a given target word. Another related task for TOWE is opinion target extraction (OTE) that attempts to identify the target words in the sentences (Qiu et al., 2011; Liu et al., 2015; Poria et al., 2016; Yin et al., 2016; Xu et al., 2018). Note that some previous works have also attempted to jointly predict the target and opinion words (Qiu et al., 2011; Liu et al., 2013; Wang et al., 2016, 2017; Li and Lam, 2017); however, the target words are still not paired with their corresponding opinion words in these studies.

As mentioned in the introduction, among a few previous work on TOWE, the main approaches include the rule-based methods (i.e., based on word distances or syntactic patterns) (Zhuang et al., 2006; Hu and Liu, 2004) and the recent deep learning models (Fan et al., 2019; Wu et al., 2020). Our model is different from the previous deep learning models as we exploit the syntactic information (i.e., dependency trees) for TOWE with deep learning.

3 Model

The TOWE problem can be formulated as a sequence labeling task. Formally, given a sentence W of N words: $W = w_1, w_2, \ldots, w_N$ with w_t as the target word ($1 \leq t \leq N$), the goal is to assign a label l_i to each word w_i so the label sequence $L = l_1, l_2, ..., l_N$ for W can capture the target-oriented opinion words for w_t. Following the previous work (Fan et al., 2019), we use the BIO tagging schema to encode the label l_i for TOWE (i.e., $l_i \in \{B, I, O\}$ for being at the **B**eginning, **I**nside or **O**utside of the opinion words respectively). Our model for TOWE consists of four components that would be described in the following: (i) Sentence Encoding, (ii) Syntax-Model Consistency, (iii) Graph Convolutional Neural Networks, and (iv) Representation Regularization.

3.1 Sentence Encoding

In order to represent the input sentence W, we encode each word w_i into a real-valued vector x_i based on the concatenation of the two following vectors: (1) the hidden vector of the first word-piece of w_i from the last layer of the BERT$_{base}$ model (Devlin et al., 2019), and (2) the position embedding for w_i. For this vector, we first compute the relative distance d_i from w_i to the target word w_t (i.e., $r_i = i - t$). Afterward, we retrieve the position embedding for w_i by looking up r_i in a position embedding table (initialized randomly). The position embeddings are fine-tuned during training in this work. The resulting vector sequence $X = x_1, x_2, \ldots, x_N$ for W will be then sent to the next computation step.

3.2 Syntax-Model Consistency

As presented in the introduction, the goal of this component is to employ the dependency tree of W to obtain the syntax-based opinion possibility scores for the words. These scores would be used to guide the representation learning of the models via the consistency with the model-based possibility scores. In particular, as we consider the closer words to the target word w_t in the dependency tree of W as being more likely to be the target-oriented opinion words, we first compute the distance d_i^{syn} between each word w_i to the target word w_t in the dependency tree (i.e., the number of words along the shortest path between w_i and w_t). Afterward, we obtain the syntax-based possibility score s_i^{syn} for w_i based on: $s_i^{syn} = \frac{\exp(-d_i^{syn})}{\sum_{j=1..N} \exp(-d_j^{syn})}$.

In order to implement the possibility score consistency, our deep learning model needs to produce $s_1^{syn}, s_2^{syn}, \ldots, s_N^{syn}$ as the model-based possibility scores the words $w_1, w_2, \ldots, w_N$ in W respectively. While the model-based score computation would be explained later, given the model-based scores, the syntax-model consistency for possibility scores would be enforced by introducing the KL divergence L_{const} between the syntax-based and model-based scores into the overall loss function to minimize:

$$L_{KL} = -\sum_i s_i^{model} \frac{s_i^{model}}{s_i^{syn}} \quad (1)$$

As mentioned in the introduction, in this work, we propose to obtain the model-based possibility scores for TOWE using the Ordered-Neuron Long Short-Term Memory Networks (ON-LSTM) (Shen et al., 2019). ON-LSTM is an extension of the popular Long Short-Term Memory Networks (LSTM) that have been used extensively in Natural Language Processing (NLP). Concretely, given the vector sequence $X = x_1, x_2, \ldots, x_N$ as the input, a LSTM layer would produce a sequence of hidden vectors $H = h_1, h_2, \ldots, h_N$ via:

$$\begin{aligned} f_i &= \sigma(W_f x_i + U_f h_{i-1} + b_f) \\ i_i &= \sigma(W_i x_i + U_i h_{i-1} + b_i) \\ o_i &= \sigma(W_o x_i + U_o h_{i-1} + b_o) \\ \hat{c}_i &= tanh(W_c x_i + U_c h_{i-1} + b_c) \\ c_i &= f_i \circ c_{i-1} + i_i \circ \hat{c}_i, h_i = o_i \circ tanh(c_i) \end{aligned} \quad (2)$$

in which h_0 is set to zero vector, $\circ$ is the element-wise multiplication, and f_t, i_t and o_t are called the forget, input, and output gates respectively.

A major problem with the LSTM cell is that all the dimensions/neurons of the hidden vectors (for the gates) are equally important as these neurons are active/used for all the step/word i in W. In other words, the words in W have the same permission to access to all the available neurons in the hidden vectors of the gates in LSTM. This might not be desirable as given a NLP task, the words in a sentence might have different levels of contextual contribution/information for solving the task. It thus suggests a mechanism where the words in the sentences have different access to the neurons in the hidden vectors depending on their informativeness. To this end, ON-LSTM introduces two additional gates $\bar{f}_i$ and $\bar{i}_i$ (the master forget and input gates) into the original LSTM mechanism using the $cummax$ activation function (i.e., $cumax(x) = cumsum(softmax(x))$)[1]:

$$\begin{aligned}
\hat{f}_i &= cummax(W_{\hat{f}}x_i + U_{\hat{f}}h_{i-1} + b_{\hat{f}}) \\
\hat{i}_i &= 1 - cummax(W_{\hat{i}}x_i + U_{\hat{i}}h_{t-1} + b_{\hat{i}}) \\
\bar{f}_i &= \hat{f}_i \circ (f_i\hat{i}_i + 1 - \hat{i}_i), \bar{i}_i = \hat{i}_i \circ (i_t\hat{f}_i + 1 - \hat{f}_i) \\
c_i &= \bar{f}_i \circ c_{i-1} + \bar{i}_i \circ \hat{c}_i
\end{aligned} \tag{3}$$

The benefit of $cummax$ is to introduce a hierarchy over the neurons in the hidden vectors of the master gates so the higher-ranking neurons would be active for more words in the sentence and vice verse (i.e., the activity of the neurons is limited to only a portion of the words in the sentence in this case). In particular, as $cummax$ applies the softmax function on the input vector whose outputs are aggregated over the dimensions, the result of $cummax(x)$ represents the expectation of a binary vector of the form $(0, \ldots, 0, 1, \ldots, 1)$ (i.e., two consecutive segments of 0's and 1's). The 1's segment in this binary vector determines the neurons/dimensions activated for the current step/word w_i, thus enabling the different access of the words to the neurons. In ON-LSTM, a word is considered as more informative or important for the task if it has more active neurons (or a larger size for its 1's segment) in the master gates' hidden vectors than the other words in the sentence. As such, ON-LSTM introduces a mechanism to estimate an informativeness score s_i^{imp} for each word w_i in the sentence based on the number of active neurons in the master gates. Following (Shen et al., 2019), we approximate s_i^{imp} via the sum of the weights of the neurons in the master forget gates, i.e., $s_i^{imp} = 1 - \sum_{j=1..D} \hat{f}_{ij}$. Here, D is the number of dimensions/neurons in the hidden vectors of the ON-LSTM gates and $\hat{f}_{ij}$ is the weight of the j-th dimension for the master forget gate $\hat{f}_i$ at w_i.

An important property of the target-oriented opinion words in our TOWE problem is that they tend to be more informative than the other words in the sentence (i.e., for understanding the sentiment of the target words). To this end, we propose to compute the model-based opinion possibility scores s_i^{model} for w_i based on the informativeness scores s_i^{imp} from ON-LSTM via: $s_i^{model} = \frac{\exp(s_i^{imp})}{\sum_{j=1..N}\exp(s_j^{imp})}$. Consequently, by promoting the syntax-model consistency as in Equation 1, we expect that the syntactic information from the syntax-based possibility scores can directly interfere with the internal computation/structure of the ON-LSTM cell (via the neurons of the master gates) to potentially produce better representation vectors for TOWE. For convenience, we also use $H = h_1, h_2, \ldots, h_N$ to denote the hidden vectors returned by running ON-LSTM over the input sequence vector X in the following.

3.3 Graph Convolutional Networks

This component seeks to extract effective context words to further improve the representation vectors H for the words in W based on the dependency connections between the words for TOWE. As discussed in the introduction, given the current word $w_i \in W$, there are two groups of important context words in W that should be explicitly encoded in the representation vector for w_i to enable effective opinion word prediction: (i) the neighboring words of w_i, and (ii) the neighboring words of the target word w_t in the dependency tree (i.e., these words should receive higher weights than the others in the representation computation for w_i). Consequently, in order to capture such important context words for all the words in the sentence for TOWE, we propose to obtain two importance score matrices of size $N \times N$ for which the scores at cells (i, j) are expected to weight the importance of the contextual information from w_j with respect to the representation vector computation for w_i in W. In particular, one score matrix would be used to capture the syntactic neighboring words of the current words (i.e., w_i) while the other score matrix would

[1] $cumsum(u_1, u_2, \ldots, u_n) = (u'_1, u'_2, \ldots, u'_n)$ where $u'_i = \sum_{j=1..i} u_j$.

be reserved for the neighboring words of the target word w_t. These two matrices would then be combined and consumed by a GCN model (Kipf and Welling, 2017) for representation learning.

Specifically, for the syntactic neighbors of the current words, following the previous GCN models for NLP (Marcheggiani and Titov, 2017; Nguyen and Grishman, 2018; Veyseh et al., 2019), we directly use the adjacency binary matrix $A^d = \{a^d_{i,j}\}_{i,j=1..N}$ of the dependency tree for W as the importance score matrix for this group of words. Note that $a^d_{i,j}$ is only set to 1 if w_i is directly connected to w_j in the dependency tree or $i = j$ in this case. In the next step for the neighboring words of the target word w_t, as we expect the closer words to the target word w_t to have larger contributions for the representation vectors of the words in W for TOWE, we propose to use the syntactic distances (to the target word) d^{syn}_i and d^{syn}_j of w_i and w_j as the features to learn the importance score matrix $A^t = \{a^t_{i,j}\}_{i,j=1..N}$ for the words in this case. In particular, $a^t_{i,j}$ would be computed by: $a^t_{i,j} = \sigma(FF([d^{syn}_i, d^{syn}_j, d^{syn}_i + d^{syn}_j, |d^{syn}_i - d^{syn}_j|, d^{syn}_i * d^{syn}_j]))$ where FF is a feed-forward network to convert a vector input with five dimensions into a scalar score and σ is the *sigmoid* function. Given the importance score matrices A^d and A^t, we seek to integrate them into a single importance score matrix A to simultaneously capture the two groups of important context words for representation learning in TOWE via the weighted sum: $A = \gamma A^d + (1 - \gamma)A^t = \{a_{i,j}\}_{i,j=1..N}$ where γ is a trade-off parameter[2].

In the next step for this component, we run a GCN model over the ON-LSTM hidden vectors H to learn more abstract representation vectors for the words in W. This step will leverage A as the adjacency matrix to enrich the representation vector for each word w_i with the information from its effective context words (i.e., the syntactic neighboring words of w_i and w_t), potentially improving the opinion word prediction for w_i. In particular, the GCN model in this work involves several layers (i.e., G layers in our case). The representation vector $\bar{h}^k_i$ for the word w_i at the k-the layer of the GCN model would be computed by:

$$\bar{h}^k_i = ReLU\left(\frac{\sum_{j=1..N} a_{i,j}(W_k \bar{h}^{k-1}_j + b_k)}{\sum_{j=1..N} a_{i,j}}\right) \quad (4)$$

where W_k and b_k are the weight matrix and bias for the k-th GCN layer. The input vector h^0_i for GCN is set to the hidden vector h_i from ON-LSTM (i.e., $h^0_i = h_i$) for all i in this case. For convenience, we denote $\bar{h}_i$ as the hidden vector for w_i in the last layer of GCN (i.e., $\bar{h}_i = \bar{h}^G_i$ for all $1 \leq i \leq N$). We also write $\bar{h}_1, \bar{h}_2, \ldots, \bar{h}_N = GCN(H, A)$ to indicate that $\bar{h}_1, \bar{h}_2, \ldots, \bar{h}_N$ are the hidden vectors in the last layer of the GCN model run over the input H and the adjacency matrix A for simplicity.

Finally, given the syntax-enriched representation vectors h_i from ON-LSTM and $\bar{h}_i$ from the last layer of GCN, we form the vector $V_i = [h_i, \bar{h}_i]$ to serve as the feature to perform opinion word prediction for w_i. In particular, V_i would be sent to a two-layer feed-forward network with the softmax function in the end to produce a probability distribution $P(.|W, t, i)$ over the possible opinion labels for w_i (i.e., B, I, and O). The negative log-likelihood function L_{pred} would then be used as the objective function to train the overall model: $L_{pred} = -\sum_{i=1}^{N} P(l_i|W, t, i)$.

3.4 Representation Regularization

There are three groups of words in the input sentence W for our TOWE problem, i.e., the target word w_t, the target-oriented opinion words (i.e., the words we want to identify) (called $W^{opinion}$), and the other words (called W^{other}). After the input sentence W has been processed by several abstraction layers (i.e., ON-LSTM and GCN), we expect that the resulting representation vectors for the target word and the target-oriented opinion words would capture the sentiment polarity information for the target word while the representation vectors for the other words might encode some other context information in W. We thus argue that the representation vector for the target word should be more similar to the representations for the words in $W^{opinion}$ (in term of the sentiment polarity) than those for W^{other}. To this end, we introduce an explicit loss term to encourage such representation distinction between these groups of words to potentially promote better representation vectors for TOWE. In particular, let R^{tar}, R^{opn}, and R^{oth} be some representation vectors for the target word w_t, the target-oriented opinion words (i.e., $W^{opinion}$),

[2]Note that we tried to directly learn A from the available information from A^d and A^t (i.e., $a_{i,j} = \sigma(FF([a^d_{i,j}, d^{syn}_i, d^{syn}_j, d^{syn}_i + d^{syn}_j, |d^{syn}_i - d^{syn}_j|, d^{syn}_i * d^{syn}_j]))$). However, the performance of this model was worse than the linear combination of A^d and A^t in our experiments.

and the other words (i.e., W^{other}) in W. The loss term for the representation distinction based on our intuition (i.e., to encourage R^{tar} to be more similar to R^{opn} than R^{oth}) can be captured via the following triplet loss for minimization:

$$L_{reg} = 1 - cosine(R^{tar}, R^{opn}) + cosine(R^{tar}, R^{oth}) \quad (5)$$

In this work, the representation vector for the target word is simply taken from last GCN layer, i.e., $R^{tar} = \bar{h}_t$. However, as $W^{opinion}$ and W^{other} might involve sets of words, we need to aggregate the representation vectors for the individual words in these sets to produce the single representation vectors R^{opn} and R^{oth}. The simple and popular aggregation method in this case involves performing the max-pooling operation over the representation vectors (i.e., from GCN) for the individual words in each set (i.e., our baseline). However, this approach ignores the structures/orders of the individual words in $W^{opinion}$ and W^{other}, and fails to recognize the target word for better customized representation for regularization. To this end, we propose to preserve the syntactic structures among the words in $W^{opinion}$ and W^{other} in the representation computation for regularization for these sets. This is done by generating the target-oriented pruned trees from the original dependency tree for W that are customized for the words in $W^{opinion}$ and W^{other}. These pruned trees would then be consumed by the GCN model in the previous section to produce the representation vectors for $W^{opinion}$ and W^{other} in this part. In particular, we obtain the pruned tree for the target-oriented opinion words $W^{opinion}$ by forming the adjacency matrix $A^{opinion} = \{a^{opinion}_{i,j}\}_{i,j=1..N}$ where $a^{opinion}_{i,j} = a_{i,j}$ if both w_i and w_j belong to some shortest dependency paths between w_t and some words in $W^{opinion}$, and 0 otherwise. This helps to maintain the syntactic structures of the words in $W^{opinion}$ and also introduce the target word w_t as the center of the pruned tree for representation learning. We apply the similar procedure to obtain the adjacency matrix $A^{other} = \{a^{other}_{i,j}\}_{i,j=1..N}$ for the pruned tree for W^{other}. Given the two adjacency matrices for the pruned trees, the GCN model in the previous section is run over the ON-LSTM vectors H, resulting in two sequences of hidden vectors for $W^{opinion}$ and W^{other}, i.e., $h'_1, h'_2, \ldots, h'_N = GCN(H, A^{opinion})$ and $h''_1, h''_2, \ldots, h''_N = GCN(H, A^{other})$. Afterward, we compute the representation vectors R^{opn} and R^{oth} for the sets $W^{opinion}$ and W^{other} by retrieving the hidden vectors for the target word returned by the GCN model with the corresponding adjacency matrices, i.e., $R^{opn} = h'_t$ and $R^{oth} = h''_t$. Note that the application of GCN over the pruned trees and the ON-LSTM vectors makes R^{opn} and R^{oth} more comparable with R^{tar} in our case. This completes the description for the representation regularizer in this work. The overall loss function in this work would be: $L = L_{pred} + \alpha L_{KL} + \beta L_{reg}$ where α and β are the trade-off parameters.

4 Experiments

4.1 Datasets & Parameters

We use four benchmark datasets presented in (Fan et al., 2019) to evaluate the effectiveness of the proposed TOWE model. These datasets contain reviews for restaurants (i.e., the datasets **14res**, **15res** and **16res**) and laptops, (i.e., the dataset **14lap**). They are created from the widely used ABSA datasets from the SemEval challenges (i.e., SemEval 2014 Task 4 (14res and 14lap), SemEval 2015 Task 12 (15res) and SemEval 2016 Task 5 (16res)). Each example in these datasets involves a target word in a sentence where the opinion words have been manually annotated.

As none of the datasets provides the development data, for each dataset, we sample 20% of the training instances for the development sets. Note that we use the same samples for the development data as in (Fan et al., 2019) to achieve a fair comparison. We use the 14res development set for hyper-parameter fine-tuning, leading to the following values for the proposed model (used for all the datasets): 30 dimensions for the position embeddings, 200 dimensions for the layers of the feed-forward networks and GCN (with $G = 2$ layers), 300 hidden units for one layer of ON-LSTM, 0.2 for γ in A, and 0.1 for the parameters α and β.

4.2 Comparing to the State of the Art

We compare the TOWE model in this work (called **ONG** for ON-LSTM and GCN) with the recent models in (Fan et al., 2019; Wu et al., 2020) and their baselines. More specifically, the following baselines are considered in our experiments:

1. **Rule-based**: These baselines employ predefined patterns to extract the opinion-target pairs that could be either **dependency-based** (Zhuang et al., 2006) or **distance-based** (Hu and Liu, 2004).

Model	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Distance-rule (2004)	58.39	43.59	49.92	50.13	33.86	40.42	54.12	39.96	45.97	61.90	44.57	51.83
Dependency-rule (2006)	64.57	52.72	58.04	45.09	31.57	37.14	65.49	48.88	55.98	76.03	56.19	64.62
LSTM (2015)	52.64	65.47	58.34	55.71	57.53	56.52	57.27	60.69	58.93	62.46	68.72	65.33
BiLSTM (2015)	58.34	61.73	59.95	64.52	61.45	62.71	60.46	63.65	62.00	68.68	70.51	69.57
Pipeline (2019)	77.72	62.33	69.18	72.58	56.97	63.83	74.75	60.65	66.97	81.46	67.81	74.01
TC-BiLSTM (2019)	67.65	67.67	67.61	62.45	60.14	61.21	66.06	60.16	62.94	73.46	72.88	73.10
IOG (2019)	82.85	77.38	80.02	73.24	69.63	71.35	76.06	70.71	73.25	82.25	78.51	81.69
LOTN (Wu et al., 2020)	84.00	80.52	82.21	77.08	67.62	72.02	76.61	70.29	73.29	86.57	80.89	83.62
ONG (Ours)	83.23	81.46	**82.33**	73.87	77.78	**75.77**	76.63	81.14	**78.81**	87.72	84.38	**86.01**

Table 1: Test set performance (i.e., Precision (P), Recall (R) and F1 scores) of the models.

2. **Sequence-based Deep Learning**: These approaches apply some deep learning model over the input sentences following the sequential order of the words to predict the opinion words (i.e., **LSTM/BiLSTM** (Liu et al., 2015), **TC-BiLSTM** (Fan et al., 2019) and **IOG** (Fan et al., 2019)).

3. **Pipeline with Deep Learning**: This method utilizes a recurrent neural network to predict the opinion words. The distance-based rules are then introduced to select the target-oriented opinion words (i.e., **Pipeline**) (Fan et al., 2019).

4. **Multitask Learning**: These methods seek to jointly solve TOWE and another related task (i.e., sentiment classification). In particular, the **LOTN** model in (Wu et al., 2020) uses a pre-trained SA model to obtain an auxiliary label for each word in the sentence using distance-based rules. A bidirectional LSTM model is then trained to make prediction for both TOWE and the auxiliary labels[3].

Table 1 shows the performance of the models on the test sets of the four datasets. It is clear from the table that the proposed ONG model outperforms all the other baseline methods in this work. The performance gap between ONG and the other models are large and significant (with $p < 0.01$) over all the four benchmark datasets (except for LOTN on 14res), clearly testifying to the effectiveness of the proposed model for TOWE. Among different factors, we attribute this better performance of ONG to the use of syntactic information (i.e., the dependency trees) to guide the representation learning of the models (i.e., with ON-LSTM and GCN) that is not considered in the previous deep learning models for TOWE.

[3]Note that (Peng et al., 2020) also proposes a related model for TOWE based on multitask deep learning. However, the models in this work actually predict general opinion words that are not necessary tied to any target word. As we focus on target-oriented opinion words, the models in (Peng et al., 2020) are not comparable with us.

4.3 Model Analysis and Ablation Study

There are three main components in the proposed ONG model, including the ON-LSTM component, the GCN component and the representation regularization component. This section studies different variations and ablated versions of such components to highlight their importance for ONG.

ON-LSTM: First, we evaluate the following variations for the ON-LSTM component: (i) **ONG - KL**: this model is similar to ONG, except that the syntax-model consistency loss based on KL L_{KL} is not included in the overall loss function, (ii) **ONG - ON-LSTM**: this model completely removes the ON-LSTM component in ONG (so the KL-based syntax-model consistency loss is not used and the input vector sequence X is directly sent to the GCN model), and (iii) **ONG_wLSTM**: this model replaces the ON-LSTM model with the traditional LSTM model in ONG (so the syntax-model consistency loss is also not employed in this case as LSTM does not support the neuron hierarchy for model-based possibility scores). The performance for these models on the test sets (i.e., F1 scores) are presented in Table 2.

Model	14res	14lap	15res	16res
ONG	82.33	75.77	78.81	86.01
ONG - KL	80.91	73.34	76.21	83.78
ONG - ON-LSTM	78.99	70.28	71.39	81.13
ONG_wLSTM	81.03	73.98	74.43	82.81

Table 2: Performance of the ON-LSTM's variations.

As we can see from the table, the syntax-model consistency loss with KL divergence is important for ONG as removing it would significantly hurt the model's performance on different datasets. The model also becomes significantly worse when the ON-LSTM component is eliminated or replaced by the LSTM model. These evidences altogether confirm the benefits of the ON-LSTM model with the

syntax-model consistency proposed in this work.

GCN Structures: There are two types of importance score matrices in the GCN model, i.e., the adjacency binary matrices A^d for the syntactic neighbors of the current words and A^t for the syntactic neighbors of the target word. This part evaluates the effectiveness of these score matrices by removing each of them from the GCN model, leading to the two ablated models **ONG -** A^d and **ONG -** A^t for evaluation. Table 3 provides the performance on the test sets for these models (i.e., F1 scores). It is clear from the table that the absence of any importance score matrices (i.e., A^d or A^t) would decrease the performance over all the four datasets and both matrices are necessary for ONG to achieve its highest performance.

Model	14res	14lap	15res	16res
ONG	82.33	75.77	78.81	86.01
ONG - A^d	80.98	73.05	75.51	83.72
ONG - A^t	81.23	74.18	76.32	85.20

Table 3: Ablation study on the GCN structures.

GCN and Representation Regularization: As the representation regularization component relies on the GCN model to obtain the representation vectors, we jointly perform analysis for the GCN and representation regularization components in this part. In particular, we consider the following variations for these two components: (i) **ONG - REG**: this model is similar to ONG except that the representation regularization loss L_{reg} is not applied in the overall loss function, (ii) **ONG_REG_wMP-GCN**: this is also similar to ONG; however, it does not apply the GCN model to compute the representation vectors R^{opn} and R^{oth} for regularization. Instead, it uses the simple max-pooling operation over the GCN-produced vectors $\bar{h}_1, \bar{h}_2, \ldots, \bar{h}_N$ of the target-oriented words $W^{opinion}$ and the other words W^{other} for R^{opn} and R^{oth}: $R^{opn} = max\_pool(\bar{h}_i | w_i \in W^{opinion})$ and $R^{oth} = max\_pool(\bar{h}_i | w_i \in W^{other})$, (iii) **ONG - GCN**: this model eliminates the GCN model from ONG, but still applies the representation regularization over the representation vectors obtained from the ON-LSTM hidden vectors. In particular, the ON-LSTM hidden vectors $H = h_1, h_2, \ldots, h_N$ would be employed for both opinion word prediction (i.e., $V = [h_i]$ only) and the computation of R^{target}, R^{opn} and R^{oth} for representation regularization with max-pooling (i.e., $R^{target} = h_t$, $R^{opn} = max\_pool(h_i | w_i \in W^{opinion})$ and $R^{oth} = max\_pool(h_i | w_i \in W^{other})$) in this case, and (iv) **ONG - GCN - REG**: this model completely excludes both the GCN and the representation regularization models from ONG (so the ON-LSTM hidden vectors $H = h_1, h_2, \ldots, h_N$ are used directly for opinion word prediction (i.e., $V = [h_i]$ as in ONG - GCN) and the regularization loss L_{reg} is not included in the overall loss function). Table 4 shows the performance of the models on the test datasets (i.e., F1 scores).

Model	14res	14lap	15res	16res
ONG	82.33	75.77	78.81	86.01
ONG - REG	80.88	73.89	75.92	84.03
ONG_REG_wMP-GCN	80.72	72.44	74.28	84.29
ONG - GCN	81.01	70.88	72.98	82.58
ONG - GCN - REG	79.23	71.04	72.53	82.13

Table 4: Performance of the variations of the GCN and representation regularization components.

There are several important observations from this table. First, as ONG - REG is significantly worse than the full model ONG over different datasets, it demonstrates the benefits of the representation regularization component in this work. Second, the better performance of ONG over ONG_REG_wMP-GCN (also over all the four datasets) highlights the advantages of the GCN-based representation vectors R^{opn} and R^{oth} over the max-pooled vectors for representation regularization. We attribute this to the ability of ONG to exploit the syntactic structures among the words in $W^{opinion}$ and W^{other} for regularization in this case. Finally, we also see that the GCN model is crucial for the operation of the proposed model as removing it significantly degrades ONG's performance (whether the representation regularization is used (i.e., in ONG - GCN) or not (i.e., in ONG - GCN - REG). The performance become the worst when both the GCN and the regularization components are eliminated in ONG, eventually confirming the effectiveness of our model for TOWE in this work.

Regularization Analysis: This section aims to further investigate the effect of the dependency structures R^{opn} and R^{oth} (i.e., among the words in $W^{opinion}$ and W^{other}) to gain a better insight into their importance for the representation regularization in this work. Concretely, we again compare the performance of the full proposed model ONG (with the graph-based representations for R^{opn} and R^{oth}) and the baseline model ONG_REG_wMP-GCN (with the direct max-pooling over the word representations,

Distance	14res		14lap	
	ONG	ONG_REG _wMP-GCN	ONG	ONG_REG _wMP-GCN
1	83.22	79.94	76.91	75.21
2	83.18	78.43	75.03	73.12
3	81.56	75.41	74.21	70.69
>3	80.97	73.77	73.92	66.23
Distance	15res		16res	
	ONG	ONG_REG _wMP-GCN	ONG	ONG_REG _wMP-GCN
1	79.92	74.29	86.52	83.33
2	78.04	73.33	87.31	83.27
3	77.71	70.91	84.77	78.63
>3	76.98	68.88	84.05	77.13

Table 5: The performance (i.e., F1 scores) of ONG and ONG_REG_wMP-GCN on the four data folds of the development sets for 14res, 14lap, 15res, and 16res. The data folds are based on the target-opinion distances of the examples (called Distance in this table).

i.e., $R^{opn} = max\_pool(\bar{h}_i|w_i \in W^{opinion})$ and $R^{oth} = max\_pool(\bar{h}_i|w_i \in W^{other})$). However, in this analysis, we further divide the sentences in the development sets into four folds and observe the models' performance on those fold. As such, for each sentence, we rely on the longest distance between the target word and some target-oriented opinion word in $W^{opinion}$ in the dependency tree to perform this data split (called the target-opinion distance). In particular, the four data folds for the development sets (of each dataset) correspond to the sentences with the target-opinion distances of 1, 2, 3 or greater than 3. Intuitively, the higher target-opinion distances amount to more complicated dependency structures among the target-oriented opinion word in $W^{opinion}$ (as more words are involved in the structures). The four data folds are thus ordered in the increasing complexity levels of the dependency structures in $W^{opinion}$.

Table 5 presents the performance of the models on the four data folds for the development sets of the datasets in this work. First, it is clear from the table that ONG significantly outperforms the baseline model ONG_REG_wMP-GCN over all the datasets and structure complexity levels of $W^{opinion}$. Second, we see that as the structure complexity (i.e., the target-opinion distance) increases, the performance of both ONG and ONG_REG_wMP-GCN decreases, demonstrating the more challenges presented by the sentences with more complicated dependency structures in $W^{opinion}$ for TOWE. However, comparing ONG and ONG_REG_wMP-GCN, we find that ONG's performance decreases slower than those for ONG_REG_wMP-GCN when the target-opinion distance increases (for all the four datasets considered in this work). This implies that the complicated dependency structures in $W^{opinion}$ have more detrimental effect on the model's performance for ONG_REG_wMP-GCN than those for ONG, leading to the larger performance gaps between ONG and ONG_REG_wMP-GCN. Overall, these evidences suggest that the sentences with complicated dependency structures for the words in $W^{opinion}$ are more challenging for the TOWE models and modeling such dependency structures to compute the representation vectors R^{opn} and R^{oth} for regularization (as in ONG) can help the models to better perform on these cases.

5 Conclusion

We propose a novel deep learning model for TOWE that seeks to incorporate the syntactic structures of the sentences into the model computation. Two types of syntactic information are introduced in this work, i.e., the syntax-based possibility scores for words (integrated with the ON-LSTM model) and the syntactic connections between the words (applied with the GCN model with novel adjacency matrices). We also present a novel inductive bias to improve the model, leveraging the representation distinction between the words in TOWE. Comprehensive analysis is done to demonstrate the effectiveness of the proposed model over four datasets.

Acknowledgement

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Zhifang Fan, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *NAACL-HLT*.

Su Su Htay and Khin Thidar Lynn. 2013. Extracting product features and opinion words using pattern knowledge in customer reviews. *The Scientific World Journal*, 2013.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *EMNLP*.

Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: a near complete solution for aspect-based sentiment analysis. In *AAAI*.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27.

Ivan Shamshurin. 2012. Extracting domain-specific opinion words for sentiment analysis. In *Mexican International Conference on Artificial Intelligence*, pages 58–68.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *ICLR*.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective lstms for target-dependent sentiment classification. In *COLING*.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *ACL*.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Quan Hung Tran, Dejing Dou, and Thien Huu Nguyen. 2020. Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation. In *EMNLP (Findings)*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*.

Zhen Wu, Fei Zhao, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2020. Latent opinions transfer network for target-oriented opinion words extraction. *arXiv preprint arXiv:2001.01989*.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *ACL*.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *IJCAI*.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50.