Effective Token Graph Modeling using a Novel Labeling Strategy for Structured Sentiment Analysis

Anonymous ACL submission

Abstract

The state-of-the-art model for structured sentiment analysis casts the task as a dependency parsing problem, which has some limitations: (1) The label proportions for span prediction and span relation prediction are imbalanced; (2) Two nodes in a dependency graph cannot have multiple arcs, which are necessary for this task; (3) The losses of predicting the imbalanced labels are directly applied in the prediction layer, which further exacerbates the imbalance problem. In this work, we propose nichetargeting solutions for these issues. First, we introduce a novel labeling strategy, which contains two sets of token pair labels, namely essential labels and whole labels. The essential label set consists of the minimum labels for this task, which are relatively balanced and applied in the prediction layer. The whole label set includes rich labels to help our model capture various token relations, which are imbalanced but merely applied in the hidden layer to softly influence our model. Moreover, we also propose an effective model to well collaborate with our labeling strategy, which is equipped with the graph attention network to iteratively refine token representations, and the adaptive multi-label classification to dynamically predict multiple relations between token pairs. We perform extensive experiments on 5 benchmark datasets in four languages. Experimental results show that our model outperforms previous SOTA models by a large margin. We believe that our labeling strategy and model can be well extended to other structured prediction tasks.

1 Introduction

Structured Sentiment Analysis (SSA), which aims to predict a structured sentiment graph as shown in Figure 1, can be formulated into the problem of tuple extraction, where a tuple (h, t, e, p) denotes a holder h who expressed an expression e towards a target t with a polarity p. Most of the existing work on sentiment analysis only focus on part of the task, such as the task of *Opinion Mining* (Katiyar and Cardie, 2016; Xia et al., 2021) which ignores the polarity classification. Recently, Barnes et al. (2021) proposed a unified approach for SSA in which they innovatively cast the sentiment analysis task as a dependency parsing problem and jointly predicts all components of a sentiment graph.

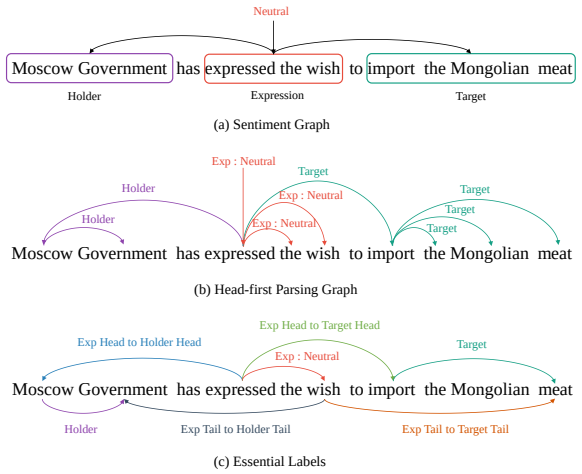

Figure 1: **(a)** An example of structured sentiment analysis. **(b)** The head-first parsing graph proposed by Barnes et al. (2021). **(c)** Our proposed *essential labels*.

However, their method may exist some problems. As seen in Figure 1(b), only 2 arcs (i.e., expressed→import and expressed→Moscow) in their parsing graph are related to span relation prediction, while much more other arcs are related to span prediction (e.g., import→the and import→meat). We argue that this imbalanced setting may hurt the extraction of the sentiment tuple, since the span lengths of sentiment tuple components (e.g., holders or targets) may be very large in this task, which will further exacerbate the label bias. Besides, the dependency parsing graph is not able to deal with multi-label classification, since it does not allow multiple arcs to share the same head and dependent tokens.

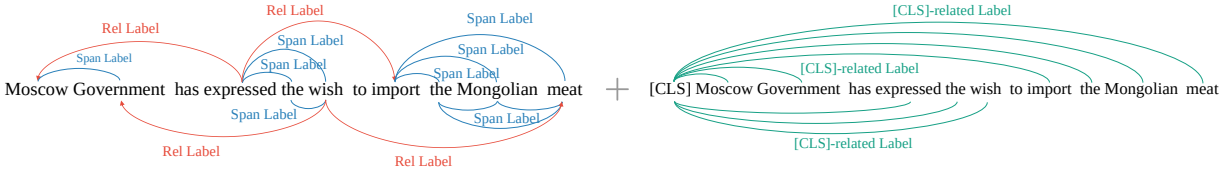

Figure 2: Whole labels contains `[CLS]`-related labels, and the labels for span prediction and span relation prediction.

To alleviate the label imbalance problem of the dependency-parsing-based method proposed by Barnes et al. (2021), we propose a novel labeling strategy that consists of two parts: First, we neglect all the labels that are related to non-boundary tokens and design a set of labels called **essential labels**, which only involves the labels that are related to boundary tokens (see Figure 1(c)).[1] As seen, the proportion of span prediction labels and span relation prediction labels are relatively balanced in the essential label set, which can mitigate the label bias problem if they are utilized in the final output layer of our model during training.

However, the labels related to non-boundary tokens of holder or target spans are also important as they can encode the relations between the tokens inside the spans, which may benefit holder, expression or target extraction with long text spans. To this end, we design another label set called **whole labels** (see Figure 2) which includes not only the labels related to boundary tokens but also the ones related to non-boundary tokens. Moreover, since the dependency-based method (Barnes et al., 2021) only considers the local relation between each pair of tokens, we add the labels between `[CLS]` and other tokens related to sentiment tuples into our whole label set, in order to utilize sentence-level global information. Considering that if the whole label set is directly applied on the output label for training, the label imbalance problem may occur again. We instead employ the whole label set in a soft and implicit fashion by applying it on the hidden layer of our model (cf. Section 4.2.2).

Based on the labeling strategy, we propose an effective token graph model, called **TGLS** (**T**oken **G**raph with a novel **L**abeling **S**trategy), to jointly predicts the label confidences for extracting all components of a sentiment tuple. First, BERT (Devlin et al., 2018) and BiLSTM are used to provide contextualized word representations. Afterwards, we built a latent graph and leverage a graph attention network (GAT) (Veličković et al., 2017) to multi-hop reason the interaction among tokens. A predictor finally classifies the essential labels between token pair and produce all possible tuples with four elements.

We conduct extensive experiments on five benchmarks, including $\text{NoReC}_{\text{Fine}}$ (Øvrelid et al., 2020), $\text{MultiB}_{\text{EU}}$, $\text{MultiB}_{\text{CA}}$ (Barnes et al., 2018), MPQA (Wiebe et al., 2005) and DS_{Unis} (Toprak et al., 2010). The resluts show that our TGLS model outperforms the current best model by a large margin. In summary, our main contributions include:

- We design a novel labeling strategy to address the label imbalance issue in prior work. Concretely, we employ the whole label set in the hidden layer to softly influence our model, and the essential label set in the prediction layer to force our model to make minimal correct predictions.
- We propose an effective graph model to well collaborate with our label strategy, which mainly includes the graph-based multi-hop reasoning to refine token representations via adjacent label edges, and the adaptive multi-label classification to dynamically adjust the decision threshold for each token pair and each label.
- The experimental results show that our model has achieved the state-of-the-art performance in 5 datasets for structured sentiment analysis, especially in terms of the end-to-end sentiment tuple extraction. Our code will be publicly available.

[1] We call them "essential" because these labels can be considered as the minimum label set that are necessary for decoding out sentiment tuples for this task.

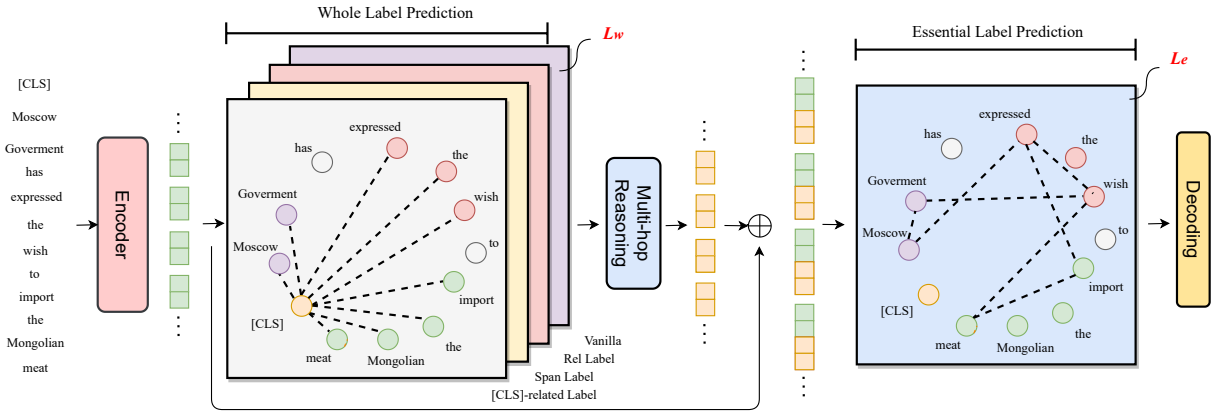

Figure 3: Overall architecture of the our framework. From left to right, the first is an encoder to yield contextualized word representations from input sentences, the next is a graph layer where we produce attention scoring matrices by whole label prediction, then follow by a multi-hop reasoning we build and refine the representation of the token, finally, a prediction layer is leveraged for reasoning the relations in essential labels and based on which we decode all components of an opinion tuple.

2 Related Works

The task of the Structured Sentiment Analysis can be divided into sub-tasks such as span extraction of the holder, target and expression, relation prediction between these elements and assigning polarity. Some existing works in *Opinion Mining* used pipeline methods to first extract spans and then the relations mostly on the MPQA dataset (Wiebe et al., 2005), such as Katiyar and Cardie (2016) propose a BiLSTM-CRF model which is the first such attempt using a deep learning approach, Zhang et al. (2019) propose a transition-based model which identifies opinion elements by the human-designed transition actions, Xia et al. (2021) propose a unified span-based model to jointly extract the span and relations. However, all of these works ignore the polarity classification sub-task.

In *End2End Aspect-Based Sentiment Analysis* (ABSA), there are also some attempts to unify several sub-tasks. Wang et al. (2016) augment the ABSA datasets with sentiment expressions, He et al. (2019) make use of this data and models the joint relations between several sub-tasks to learn common features, (Chen and Qian, 2020) also exploit interactive information from each pair of sub-tasks (target extraction, expression extraction, sentiment classification). However, Wang et al. (2016) only annotate sentiment-bearing words not phrases and do not specify the relationship between target and expression, it therefore may not be adequate for full structured sentiment analysis.

Thus, Barnes et al. (2021) propose a unified approach in which they formulate the structured sentiment analysis task into a dependency graph parsing task and jointly predicts all components of a sentiment graph. However, as aforementioned, this direct transformation may be problematic as it may introduce label imbalance in span and relation prediction. Thus, we propose an effective graph model with a novel labeling strategy in which we employ a whole label set in the hidden layer to softly affect our model, and an essential label set in the prediction layer to address the imbalance issue.

The design of our essential label set is inspired by the Handshaking Tagging Scheme (Wang et al., 2020), which is a token pair tagging scheme for entity and relation extraction. The Handshaking Tagging Scheme involves only the labels related to the boundary tokens and enables a one-stage joint extraction of spans and relations.

3 Token-Pair Labeling Scheme

3.1 Essential Labels

The design of the essential label set is inspired by Handshaking Tagging Scheme (Wang et al., 2020), which only involves the labels related to the boundary tokens. Thus, the label proportions for span prediction and span relation prediction are relatively balanced, which mitigates the label imbalance problem in prior work (Barnes et al., 2021). For essential labels, we use them in the prediction layer to decode sentiment tuples.

3.2 Whole Labels

As seen in Figure 2, the whole label set involves both the labels related to boundary and non-boundary tokens, as well as three labels related to `[CLS]` and all tokens in the sentiment tuples. Thus, our whole label set can be divided into three groups, span labels, relation labels and `[CLS]`-related labels. Non-boundary tokens make our model be aware of the relations between the inside tokens of a holder, expression or target span, and `[CLS]`-related labels help inject the sentence-level global information into our model. We apply whole labels in the hidden layer to softly embed the above information into our model, in order to avoid the potential label imbalance issue.

4 Methodology

The architecture of our framework is illustrated in Figure 3, which mainly consists of three components. First bi-directional LSTM is employed as the encoder to yield contextualized word representations from input sentences. Then a graph layer is used to build and refine the representation of the token, effectively capturing the token interaction among spans and global information with whole labels. Finally, a prediction layer is leveraged for reasoning the relations in essential labels between all word pairs.

4.1 Encoder Layer

Consider the i^{th} token in a sentence with n tokens, we represent it by concatenating its token embedding $\mathbf{e}_i^{word}$, part-of-speech (POS) embedding $\mathbf{e}_i^{pos}$, lemma embedding $\mathbf{e}_i^{lemma}$, and character-level embedding $\mathbf{e}_i^{char}$ together:

$$\boldsymbol{w}_i = \mathbf{e}_i^{word} \oplus \mathbf{e}_i^{pos} \oplus \mathbf{e}_i^{lemma} \oplus \mathbf{e}_i^{char} \quad (1)$$

where $\oplus$ denotes the concatenation operation. The character-level embedding is generated by the convolution neural networks (CNN) (Kalchbrenner et al., 2014). Then, we employ bi-directional LSTM (BiLSTM) to encode the vectorial token representations into contextualized word representations:

$$\mathbf{h}_i = \mathrm{BiLSTM}\left(\mathbf{w}_i\right) \quad (2)$$

where $\mathbf{h}_i$ is the token hidden representation.

Moreover, in the same way as previous work (Barnes et al., 2021), we also enhance token representations with pretrained contextualized embeddings using multilingual BERT (Devlin et al., 2018).

4.2 Graph Layer

4.2.1 Token Graph

We treat our graph as a latent variable, where the graph nodes are the token representations from the encoder layer, and the graph edges are formulated into the adjacency attention matrix of the graph attention network (GAT) (Veličković et al., 2017). The proposed token graph includes 4 views, where each view corresponds to an adjacency attention matrix. Recall that the whole label set is applied in this layer, which includes three groups of labels. Thus, three attention matrices are used to predict three groups of labels respectively, while one attention matrix is used without any prediction task, as the method in vanilla GAT. Formally, we represent the latent token graph $\boldsymbol{G}$ as follows:

$$\boldsymbol{G} = \left(V, S_0^{\mathcal{G}}, S_s^{\mathcal{G}}, S_r^{\mathcal{G}}, S_c^{\mathcal{G}}\right) \quad (3)$$

where V is the set of tokens, $S_0^{\mathcal{G}}$ is the attention matrix in vanilla GAT, $S_s^{\mathcal{G}}$, $S_r^{\mathcal{G}}$ and $S_c^{\mathcal{G}}$ are the attention matrices used to predict span prediction labels, span relation prediction labels and `[CLS]`-related labels respectively.

4.2.2 Whole Label Prediction

In this section, we introduce the process that whole labels influence the graph layer by label prediction using the attention scores of attention matrices $S_s^{\mathcal{G}}$, $S_r^{\mathcal{G}}$ and $S_c^{\mathcal{G}}$. Without loss of generality, we employ $S^{\mathcal{G}}$ unifiedly.

Attention Scoring Our attention matrices are produced by a mechanism of Attention Scoring which takes two token representations $\boldsymbol{h}_i, \boldsymbol{h}_j$ as the input and for the n^{th} attention matrix, we first map the tokens to $q(\boldsymbol{h}_i, n)$ and $k(\boldsymbol{h}_j, n)$ with two multi-layer perceptions (MLP):

$$\begin{aligned} q(\boldsymbol{h}_i, n) &= MLP_n^q\left(\boldsymbol{h}_i\right) \\ k(\boldsymbol{h}_j, n) &= MLP_n^k\left(\boldsymbol{h}_j\right) \end{aligned} \quad (4)$$

Then we apply a technique of Rotary Position Embedding (RoPE) (Su et al., 2021) to encode relative position information. The attention score $S_n^{\mathcal{G}}(i,j)$ can be calculated as follows:

$$\begin{aligned} S_n^{\mathcal{G}}(i,j) &= score(\boldsymbol{h}_i, \boldsymbol{h}_j, n) \\ score(\boldsymbol{h}_i, \boldsymbol{h}_j, n) &= \left(q(\boldsymbol{h}_i, n)\right)^{\top} \boldsymbol{R}_{j-i} k(\boldsymbol{h}_j, n) \end{aligned} \quad (5)$$

where $\boldsymbol{R}_{j-i}$ incorporates explicit relative positional information in attention scoring.

And in the same way as calculating $S_n^{\mathcal{G}}(i,j)$, we can produce all token pair scores of all adjacency attention matrix, thus inducing the whole graph edges $S^{\mathcal{G}}$:

$$S^{\mathcal{G}} = \left\{ S_n^{\mathcal{G}}(i,j) | 1 \leq n \leq N, 1 \leq i,j \leq l \right\} \quad (6)$$

where n denotes the n^{th} adjacency attention matrix, N is the total number of the matrix, l is the length of the sentence.

Then, we introduce an adaptive thresholding function below, which produces a token pair dependent threshold to enable the injection of the information from whole labels $\mathcal{R}_u$ into the adjacency attention matrix.

Adaptive Thresholding for a certain token pair with representations of $\boldsymbol{h}_i, \boldsymbol{h}_j$, the token pair dependent threshold and the whole $TH^{\mathcal{G}}$ are calculated as follows:

$$\begin{aligned} TH^{\mathcal{G}} &= \left\{ TH_{ij}^{\mathcal{G}} | 1 \leq i,j \leq l \right\} \\ TH_{ij}^{\mathcal{G}} &= threshold(\boldsymbol{h}_i, \boldsymbol{h}_j) \end{aligned} \quad (7)$$

where the $threshold(\boldsymbol{h}_i, \boldsymbol{h}_j)$ is defined as:

$$\begin{aligned} q^{TH}(\boldsymbol{h}_i) &= \boldsymbol{W}_q \boldsymbol{h}_i + \boldsymbol{b}_q \\ k^{TH}(\boldsymbol{h}_j) &= \boldsymbol{W}_k \boldsymbol{h}_j + \boldsymbol{b}_k \\ threshold(\boldsymbol{h}_i, \boldsymbol{h}_j) &= \left(q^{TH}(\boldsymbol{h}_i) \right)^\top \boldsymbol{R}_{j-i} k^{TH}(\boldsymbol{h}_j) \end{aligned} \quad (8)$$

where $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{b}_q$ and $\boldsymbol{b}_k$ are the trainable weight and bias matrix, $\boldsymbol{R}_{j-i}$ are calculated in the same way as Eq.(5), which is used to incorporate explicit relative positional information.

Then combined with a multi-label adaptive-threshold loss and for a certain whole label $r \in \mathcal{R}_w$ and the corresponding adjacency attention matrix $S_r^{\mathcal{G}}$, we push the logits $S_r^{\mathcal{G}}(i,j)$ above the adaptive threshold $TH_{ij}^{\mathcal{G}}$ when the token pair possesses the label, and pull below when it does not.

Due to the abundance of whole labels and the flexibility of the adaptive threshold, it allows the model to induce a more informative adjacency attention matrix for our token graph.

4.2.3 Multi-hop Reasoning

Considering that the adjacency attention matrix $S^{\mathcal{G}}$ is embedded with the information from whole labels $\mathcal{R}_w$, we naturally think of applying the multi-head graph attention networks (GATs) (Veličković et al., 2017) for multi-hop reasoning to obtain more informative token representations. Specifically, we first apply a softmax on our adjacency attention matrix $S^{\mathcal{G}}$, then the GATs computation for the representation $\boldsymbol{u}_i^{l+1}$ of the token i at the $(l+1)^{th}$layer, which takes the representations from previous layer as input and outputs the updated representations, can be defined as:

$$A = Softmax\left(S^{\mathcal{G}}\right) \quad (9)$$

$$\boldsymbol{u}_i^{l+1} = \sigma\left(\frac{1}{N} \sum_{n=1}^{N} \sum_{j \in \mathcal{N}_i^n} A_{ij}^n \boldsymbol{W}_l^n \boldsymbol{u}_j^l \right) \quad (10)$$

where $\boldsymbol{W}_l^n$ is the trainable weight matrix for l^{th} layer and n^{th} adjacency attention matrix, $\mathcal{N}_i^n$ is the neighbor of token i, σ is the ReLU (Nair and Hinton, 2010) activation function.

4.3 Prediction Layer

For each token, we get the final representation $\boldsymbol{c}_i$ by taking a shortcut connection between the outputs of Encoder Layer and Graph Layer:

$$\boldsymbol{c}_i = \boldsymbol{h}_i \oplus \boldsymbol{u}_i \quad (11)$$

To identify possible essential labels $\mathcal{R}_e$ of each token pair, we calculate the token pair score matrix $S^{\mathcal{P}}, r \in \mathcal{R}_e$ and the adaptive threshold $TH^{\mathcal{P}}$ based on the function of attention scoring and adaptive threshold (see Eq.(5) and Eq.(8)):

$$\begin{aligned} S_r^{\mathcal{P}}(i,j,r) &= score(\boldsymbol{c}_i, \boldsymbol{c}_j, r) \\ TH_{ij}^{\mathcal{P}} &= threshold(\boldsymbol{c}_i, \boldsymbol{c}_j) \\ S^{\mathcal{P}} &= \left\{ S_r^{\mathcal{P}}(i,j) | 1 \leq i,j \leq l, r \in \mathcal{R}_e \right\} \\ TH^{\mathcal{P}} &= \left\{ TH_{ij}^{\mathcal{P}} | 1 \leq i,j \leq l \right\} \end{aligned} \quad (12)$$

Formally, the essential labels for a certain token pair c_i, c_j is predicted by following equation:

$$\Omega_{ij} = \left\{ r | S_r^{\mathcal{P}}(i,j) > TH_{ij}^{\mathcal{P}}, r \in \mathcal{R}_e \right\} \quad (13)$$

where token pair satisfying $S_r^{\mathcal{P}}(i,j) > TH_{ij}^{\mathcal{P}}$ are regarded as possessing label $r \in \mathcal{R}_e$, and Ω_{ij} is the set of predicted essential labels of token pair c_i, c_j.

4.4 Training

In our work, we apply a loss[2] that extends cross entropy to multi-label classification problem. However, we replace the global threshold with a token pair dependent threshold to enable the information injection from whole labels $\mathcal{R}_w$ to adjacency

[2]The loss was proposed by Su on the blog website https://kexue.fm/archives/7359.

Dataset	Model	Span				Targeted	Sent. Graph	
		Holder F1	Target F1	Exp. F1	Avg. F1	F1	NSF1	SF1
NoReC$_{\text{Fine}}$	RACL-BERT	-	47.2	56.3	-	30.3	-	-
	Head-first	51.1	50.1	54.4	53.1*	30.5	37.0	29.5
	Head-final	60.4	**54.8**	55.5	55.7*	31.9	39.2	31.2
	TGLS	**60.9**	53.2	**61.0**	**58.1**	**38.1**	**46.4**	**37.6**
MultiB$_{\text{EU}}$	RACL-BERT	-	59.9	72.6	-	56.8	-	-
	Head-first	60.4	64.0	73.9	69.6*	57.8	58.0	54.7
	Head-final	60.5	64.0	72.1	68.2*	56.9	58.0	54.7
	TGLS	**62.8**	**65.6**	**75.2**	**71.0**	**60.9**	**61.1**	**58.9**
MultiB$_{\text{CA}}$	RACL-BERT	-	67.5	70.3	-	52.4	-	-
	Head-first	43.0	72.5	71.1	70.5*	55.0	62.0	56.8
	Head-final	37.1	71.2	67.1	70.2*	53.9	59.7	53.7
	TGLS	**47.4**	**73.8**	**71.8**	**71.6**	**60.6**	**64.2**	**59.8**
MPQA	RACL-BERT	-	20.0	31.2	-	17.8	-	-
	Head-first	43.8	51.0	**48.1**	**47.7***	**33.5**	24.5	17.4
	Head-final	**46.3**	49.5	46.0	47.2*	18.6	26.1	18.8
	TGLS	44.1	**51.7**	47.8	47.0	23.3	**28.2**	**21.6**
DS$_{\text{Unis}}$	RACL-BERT	-	44.6	38.2	-	27.3	-	-
	Head-first	28.0	39.9	40.3	40.1*	26.7	31.0	25.0
	Head-final	37.4	42.1	**45.5**	43.0*	29.6	34.3	26.5
	TGLS	**43.7**	**49.0**	42.6	**45.7**	**31.6**	**36.1**	**31.1**

Table 1: Main experimental results of our TGLS model and comparison with previous works. The baseline results with "*" are from our reimplementation, the others are from (Barnes et al., 2021).

attention matrix of the GATs. The loss is also applied in the Prediction Layer to identify all possible essential labels for each token pair to solve the multi-label problem. Formally, the multi-label adaptive-threshold loss function in prediction layer is defined as follows:

$$\begin{aligned}\mathcal{L}_e &= L(TH^{\mathcal{P}}, S^{\mathcal{P}}) \\ &= \sum_i \sum_{j>i} \log \left(e^{TH_{ij}^{\mathcal{P}}} + \sum_{r \in \Omega_{ij}^{neg}} e^{S_r^{\mathcal{P}}(i,j)} \right) \\ &+ \sum_i \sum_{j>i} \log \left(e^{-TH_{ij}^{\mathcal{P}}} + \sum_{r \in \Omega_{ij}^{pos}} e^{-S_r^{\mathcal{P}}(i,j)} \right)\end{aligned} \tag{14}$$

where Ω_{ij}^{pos} and Ω_{ij}^{neg} are positive and negative classes involving link labels that exist or not exist between token i and token j. When minimizing the loss, it pushes the logits of all positive classes above the corresponding threshold $TH_{ij}^{\mathcal{P}}$, and pulles the logits of negative classes below.

In a similar way we can get the loss $\mathcal{L}_w$ in Graph Layer by taking the $TH^{\mathcal{G}}$, $S^{\mathcal{G}}$ as the inputs of the loss function. Thus the whole loss of our model can be calculated as follows:

$$\mathcal{L}_{all} = \mathcal{L}_e + \alpha \mathcal{L}_w \tag{15}$$

where the α is a hyperparameter to adjust the ratio of the two losses.

5 Experimental Settings

5.1 Datasets and Configuration

For comparison with previous sota work (Barnes et al., 2021), we perform experiments on five structured sentiment datasets in four languages, including multi-domain professional reviews **NoReC**$_{\text{Fine}}$ (Øvrelid et al., 2020) in Norwegian, hotel reviews **MultiB**$_{\text{EU}}$ and **MultiB**$_{\text{CA}}$ (Barnes et al., 2018) in Basque and Catalan respectively, news **MPQA** (Wiebe et al., 2005) in English and reviews of online universities and e-commerce **DS**$_{\text{Unis}}$ (Toprak et al., 2010) in English.

For fair comparison, we use word2vec skip-gram embeddings openly available from the NLPL vector repository (Kutuzov et al., 2017). Our model is implemented with PyTorch and the network

weights are optimized with Adam (Kingma and Ba, 2014). We also conduct Cosine Annealing Warm Restarts learning rate schedule (Loshchilov and Hutter, 2016). We train our models for at most 100 epochs and choose the model with the best performance in SF1 score on the validation set to output results on the test set. And we run all of our models three times with different random seeds. Finally, the average results of the three runs are reported in our work (Hyper-parameter settings are listed in Table 4).

5.2 Baselines

We compare our proposed model with three state-of-the-art baselines which outperform other models in all datasets:

RACL-BERT Chen and Qian (2020) propose a relation-aware collaborative learning framework for end2end sentiment analysis which models the interactive relations between each pair of sub-tasks (target extraction, expression extraction, sentiment classification). Barnes et al. (2021) reimplement the RACL as a baseline for SSA task in their work, and they also enhance token representations using multilingual BERT (Devlin et al., 2018).

Head-first and Head-final Barnes et al. (2021) cast the structured sentiment analysis as a dependency parsing task and apply a reimplementation of the neural parser by Dozat and Manning (2018), where the main architecture of the model is based on a biaffine classifier. The Head-first and Head final are two models with different setups in the parsing graph.

5.3 Evaluation Metrics

Following previous sota work Barnes et al. (2021), we use the Span F1, Targeted F1 and two Sentiment Graph Metrics to measure the experimental results.

In detail, Span F1 evaluates how well these models are able to identify the holders, targets, and expressions. Targeted F1 requires the exact extraction of the correct target, and the corresponding polarity. Sentiment Graph Metrics include two F1 score, Non-polar Sentiment Graph F1 (NSF1) and Sentiment Graph F1 (SF1), which aims to measure the overall performance of a model to capture the full sentiment graph (see Figure 1a). For NSF1, each sentiment graph is a tuple of (holder, target, expression), while SF1 adds the polarity (holder, target, expression, polarity). A true positive is defined as an exact match at graph-level, weighting the overlap in predicted and gold spans for each element, averaged across all three spans.

Moreover, for ease of analysis, we add an Average Span F1 Score which evaluates how well these models are able to identify all three elements of a sentiment graph with token-level F1.

	NoReC$_{\text{Fine}}$	**MultiB$_{\text{EU}}$**	**MultiB$_{\text{CA}}$**	**MPQA**	**DS$_{\text{Unis}}$**
Head-final	52.3	63.9	67.3	45.0	41.5
TGLS					
+parsing labels	54.2	65.4	67.5	44.7	43.2
+essential labels	**57.8**	**68.7**	**70.1**	**46.1**	**45.7**

Table 2: Experimental results and comparison of the pure relation extraction F1 scores.

6 Results

In this section, we introduce the main experimental results (see Table 1) compared with three state-of-the-art models RACL-BERT (Chen and Qian, 2020), Head-first and Head-final models (Barnes et al., 2021).

Table 1 shows that in most cases our TGLS model performs better than other baselines in terms of the Span F1 metric on all datasets. And the average improvement (↑ 1.4) in Avg. Span F1 score proves the effectiveness of our model in span extraction. Besides, there exists some significant improvements such as extracting holder on **DS$_{\text{Unis}}$** (↑6.3) and extracting expression on **NoReC$_{\text{Fine}}$** (↑4.7), but the extracting expression on **DS$_{\text{Unis}}$** (↓2.9) are poor.

As for the metric of Targeted F1, although the Head-first model performs well on **MPQA**, our TGLS model is obviously more robust as we achieves superior performance on other 4 datasets. There are also extremely significant improvements such as on **NoReC$_{\text{Fine}}$** (↑6.2) and on **MultiB$_{\text{CA}}$** (↑5.6), it proves the capacity of our model in exact prediction of target and the corresponding polar.

As for the Sentiment Graph metrics, which are important for comprehensively examining span, relation and polar predictions, our TGLS model achieves superior performance throughout all datasets in both NSF1 and SF1 score, especially on **NoReC$_{\text{Fine}}$** (↑7.2 and ↑6.4). And the average improvement (↑4.5) in SF1 score verifies the excellent ability of our model in the end-to-end sentiment tuple extraction, which is the key point in Structured Sentiment Analysis task.

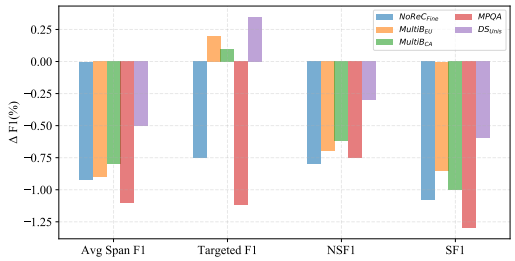

Figure 4: Performance drops without the whole labels.

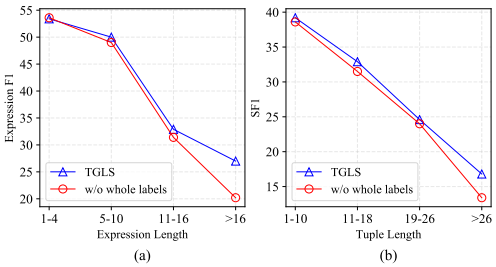

Figure 5: Analysis on the whole labels on **NoReC**$_{\text{Fine}}$. **(a)** Expression F1 score regarding to different expression length. **(b)** SF1 score regarding to different tuple length.

7 Discussion

In this section we perform a deeper analysis on the models in order to answer two research questions:

7.1 Do our model mitigates the label bias in span and relation prediction?

We hypothesize that the dependency-parsing-based method proposed by Barnes et al. (2021) may introduce the label imbalance problem and affect the efficiency in relation prediction, we therefore only use the essential labels in the prediction layer to make minimal correct predictions. Experimental results show that our model performs significantly better in the overall metric SF1, which to some extent proves that our model can simultaneously ensure the efficiency of span and relation extraction. However, it is still a worthy question to explore whether and how much do our essential labels improve the performance of relation prediction?

For ease of analysis, we replace our essential labels with the dependency-parsing-based labels (Barnes et al., 2021) in the prediction layer and experiment on all datasets in terms of a relation prediction metric, where a true positive is defined as any span pair that overlaps the gold span pair and has the same relation. Table 2 shows that our model significantly improve the performance of relation prediction compared with previous sota model (Barnes et al., 2021) on all datasets. Besides, we can see that our model with essential labels achieves superior performance than the model with replaced dependency-parsing-based labels, which proves the effectiveness of using essential labels to improve the performance of relation prediction.

7.2 Do the utilization of whole labels improve the result?

In this section, we first evaluate our model on all datasets in terms of the Avg Span F1 and Targeted F1, NSF1 and SF1 scores by directly drop the whole labels. Figure 3 shows the performance drops without the whole labels, the whole labels almost improves the performance in all metrics on all datasets, although the **MultiB**$_{\text{EU}}$, **MultiB**$_{\text{CA}}$ and **DS**$_{\text{Unis}}$ in Targeted F1 metric are exceptions, this may attributed to the three datasets have shorter targets, and it indicates that the whole labels may benefit more from long span issues.

Then, we experiment on **NoReC**$_{\text{Fine}}$ to further explore whether whole labels contribute to long span issues? Figure 5a evaluates the Expression F1 score regarding to different expression length, we can find that *whole labels helps most on those expressions with longer length*. We also report the SF1 score regarding to different distance from the token in tuple with smallest position to the token with largest position in Figure 5b, which shows a similar conclusion.

8 Conclusion

In this paper, we propose a graph model **TGLS** with a novel labeling strategy, consisting of whole labels and essential labels, to extract opinion tuples for structured sentiment analysis. By predicting whole labels, our model is capable of capturing global and token pair interaction information. We further propose a multi-hop reasoning graph layer for better refining the token representations via the latent graph built from the whole label prediction. We conduct extensive experiments on five benchmark datasets to validate the effectiveness of the proposed framework. Experimental results show that our model overwhelmingly outperforms SOTA baselines.

References

Jeremy Barnes, Toni Badia, and Patrik Lambert. 2018. MultiBooked: A corpus of Basque and Catalan hotel reviews annotated for aspect-level sentiment classification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. Structured sentiment analysis as dependency graph parsing. *arXiv preprint arXiv:2105.14504*.

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3694, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, Florence, Italy. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Arzoo Katiyar and Claire Cardie. 2016. Investigating lstms for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Icml*.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. 2010. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 575–584, Uppsala, Sweden. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626, Austin, Texas. Association for Computational Linguistics.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. *arXiv preprint arXiv:2010.13415*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, 39(2-3):165–210.

Qingrong Xia, Bo Zhang, Rui Wang, Zhenghua Li, Yue Zhang, Fei Huang, Luo Si, and Min Zhang. 2021. A unified span-based approach for opinion mining with syntactic constituents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1795–1804.

Meishan Zhang, Qiansheng Wang, and Guohong Fu. 2019. End-to-end neural opinion extraction with a transition-based model. *Information Systems*, 80:56–63.

Dataset	Head-final	TGLS
NoReC$_{\text{Fine}}$,	57	62.1 (↑5.1)
MultiB$_{\text{EU}}$,	75.7	79.3 (↑3.6)
MultiB$_{\text{CA}}$,	71.7	76.2 (↑4.5)
MPQA	38.5	41.0 (↑2.5)
DS$_{\text{Unis}}$,	44.5	47.8 (↑3.3)

Table 3: Experimental results and comparison of the Polarity F1 scores.

Hyperparameter	**Best assignment**
contexualized embedding	mBERT
embeddings trainable	FALSE
number of epochs	100
batch size	8
learning rate	3e-5
α	0.25
hidden lstm	400
layers lstm	4
dim embedding	100
dim char embedding	100
dropout embedding	0.4
dropout main recurrent	0.3

Table 4: Detailed settings of our hyper-parameter.

A Analysis of polarity predictions

In this section, we focus on the performance in only polarity prediction, where a true positive is defined as any expression that overlaps the gold expression with the same polarity. Table 3 shows that our model achieves superior performance than previous sota model (Barnes et al., 2021) on all datasets, especially on **NoReC**$_{\text{Fine}}$ (↑5.1), which has longer expressions, it once again verifies that our model has excellent performance on the long span problem.