A Unified Model for Opinion Target Extraction and Target Sentiment Prediction*

Xin Li,[1] **Lidong Bing,**[2] **Piji Li,**[3] **Wai Lam**[1]
[1]Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
[2]R&D Center Singapore, Machine Intelligence Technology,
Alibaba DAMO Academy
[3]Tencent AI Lab, Shenzhen, China
{lixin, wlam}@se.cuhk.edu.hk, l.bing@alibaba-inc.com, pijili@tencent.com

Abstract

Target-based sentiment analysis involves opinion target extraction and target sentiment classification. However, most of the existing works usually studied one of these two sub-tasks alone, which hinders their practical use. This paper aims to solve the complete task of target-based sentiment analysis in an end-to-end fashion, and presents a novel unified model which applies a unified tagging scheme. Our framework involves two stacked recurrent neural networks: The upper one predicts the unified tags to produce the final output results of the primary target-based sentiment analysis; The lower one performs an auxiliary target boundary prediction aiming at guiding the upper network to improve the performance of the primary task. To explore the inter-task dependency, we propose to explicitly model the constrained transitions from target boundaries to target sentiment polarities. We also propose to maintain the sentiment consistency within an opinion target via a gate mechanism which models the relation between the features for the current word and the previous word. We conduct extensive experiments on three benchmark datasets and our framework achieves consistently superior results.

Introduction

Target-Based Sentiment Analysis (TBSA) aims to detect the opinion targets explicitly mentioned in sentences and predict the sentiment polarities over the opinion targets (**?**; **?**). For example, in the sentence "***USB3 Peripherals** are noticably less expensive than the **ThunderBolt ones***", the user mentions two opinion targets, namely, "***USB3 Peripherals***" and "***ThunderBolt ones***", and expresses positive sentiment over the first, and negative sentiment over the second.

Traditionally, this task can be broken into two sub-tasks, namely, opinion target extraction and target sentiment classification. The goal of opinion target extraction is to detect the opinion target mentions in the text, and it has been extensively studied (**?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**). The second sub-task, i.e., target sentiment classification, performs as a multiplier for the usefulness of the extracted target mentions, as it can predict the sentiment polarity of the given opinion targets. This sub-task has also received a lot of attention in recent years (**?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**). However, most existing methods solving the second sub-task assume that the target mentions are given, which limits their practical use. To sum up, all the above works aim at solving only one of the sub-tasks. In order to apply these existing methods in practical settings, i.e., not only extracting the targets, but also predicting the target sentiment, one typical way is to pipeline the methods of the two sub-tasks together.

As observed in some other tasks (**?**; **?**; **?**; **?**), if two sub-tasks have strong couplings (e.g, NER and relation extraction), a more integrated model is usually more effective than a pipline solution. For the TBSA task, previous researchers have attempted two approaches to a more integrated solution (**?**; **?**). One approach is to make the models of the two sub-tasks jointly trained, which utilizes a set of target boundary tags (e.g., `B`, `I`, `E`, `S` and `O`) and a set of sentiment tags (e.g. `POS`, `NEG`, `NEU`). The "joint" row of Table 1 gives an example of the tagging scheme in this approach. Another approach is to totally dismiss the boundary of the two sub-tasks, which utilizes a set of specially-designed tags (we name it "unified tagging scheme"), namely, `B-{POS, NEG, NEU}`, `I-{POS, NEG, NEU}`, `E-{POS, NEG, NEU}`, `S-{POS, NEG, NEU}`, denoting the beginning of, inside of, end of, and single-word opinion target with positive, negative or neutral sentiment respectively, and `O` denoting NULL sentiment. An example is given in the "unified" row in Table 1. Unfortunately, these initial attempts did not result in a more integrated model that can outperform the pipeline approaches.

Although the importance of solving the complete TBSA task remains significant, existing studies are relatively less and their findings (**?**; **?**), to some extent, discouraged other researchers to do further explorations. However, we think that research efforts should be paid to explore a more integrated model for solving this task, because its two sub-tasks are highly coupled together and the potential of a more integrated model is promising.

In this paper, we investigate the complete task of TBSA and design a novel unified framework to handle it in an end-to-end fashion. The proposed framework involves two

*The work described in this paper was mainly done when Lidong Bing was an employee and Xin Li was an intern at Tencent AI Lab. It is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Codes: 14203414) and the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 4055093).

Input	The	AMD	Turin	Processor	seems	to	always	perform	much	better	than	Intel	.
Joint	O	B	I	E	O	O	O	O	O	O	O	S	O
	O	POS	POS	POS	O	O	O	O	O	O	O	NEG	O
Unified	O	B-POS	I-POS	E-POS	O	O	O	O	O	O	O	S-NEG	O

Table 1: Tagging schemes used in the integrated approaches. "Joint" and "Unified" refers to joint and unified approaches respectively.

stacked Recurrent Neural Networks (RNN). The upper one produces the final tagging results of the TBSA task based on the unified tagging scheme. The lower one performs an auxiliary prediction of target boundaries with the aim for guiding and providing the information to the upper RNN. Such design is based on the observation that under the unified tagging scheme, the span information is exactly identical to that under the boundary tagging scheme. Refer to the example in Table 1, if a word is at the beginning of a target mention under the boundary scheme, i.e., having the tag B, it should also be at the beginning under the unified scheme, i.e., having the tag B-POS. In order to explore such inter-scheme tag dependency, we propose to guide the prediction of the upper RNN for the complete TBSA task with the boundary prediction from the auxiliary task, corresponding to the lower RNN. Specifically, we design a component to encode the dependencies into a transition matrix and use the matrix to map the probability distribution of the boundary prediction to the unified tag space of the TBSA task. Then, we determine the proportions of the obtained boundary-based probability scores in the tagging decision and consolidate them with the probability scores from the upper RNN for final predictions.

We also propose to maintain the consistency of the sentiment of individual words within the same target mention based on a simple gate mechanism. The gate mechanism is designed to explicitly consolidate the features of the current word and the previous word. Since both of the gate here and the transition matrix above need to take reliable boundary prediction for performing well, improving the reliability of such prediction in the lower RNN is supposed to be useful for the complete TBSA task. Therefore, we introduce another component to estimate the potential of a word to be a target word. Note that as defined by the task (**?**; **?**; **?**), an opinion target should always co-occur with opinion words, thus, the words close to the opinion words are more likely to be target words and we obtain additional supervision signals for refining boundary information based on this assumption.

In the experiments, our framework outperforms the state-of-the-art methods and the strongest sequence taggers on several benchmark datasets. We conducted detailed ablation studies to quantitatively demonstrate the effectiveness of the designed components. With some case analysis, we show how our framework can handle some difficult cases with the help of the designed components.

Our Proposed Framework

Task Definition

We formulate the complete Target-Based Sentiment Analysis (TBSA) task as a sequence labeling problem, and employ a unified tagging scheme: $\mathcal{Y}^{\mathcal{S}} = \{$B-POS, I-POS, E-POS, S-POS, B-NEG, I-NEG, E-NEG, S-NEG, B-NEU, I-NEU, E-NEU, S-NEU, O$\}$. Except O, each tag contains two parts of tagging information: the boundary of target mention, and the target sentiment. For example, B-POS denotes the beginning of a positive target mention, and S-NEG denotes a single-word negative opinion target. For a given input sequence $\mathrm{X} = \{x_1, \ldots, x_T\}$ with length T, our goal is to predict a tag sequence $\mathrm{Y}^{\mathcal{S}} = \{y_1^{\mathcal{S}}, \ldots, y_T^{\mathcal{S}}\}$, where $y_i^{\mathcal{S}} \in \mathcal{Y}^{\mathcal{S}}$.

Model Description

Overview As shown in Figure 1, on the top of two stacked RNNs with LSTM cells, our framework designs three tailor-made components, depicted in detail with the callouts, to explore three important intuitions in the task of TBSA. Specifically, the upper LSTM$^{\mathcal{S}}$ is for the complete TBSA task and it predicts the unified tags as output, while the lower LSTM$^{\mathcal{T}}$ is for the auxiliary task and predicts the boundary tags of target mentions. The boundary prediction from LSTM$^{\mathcal{T}}$ is used to guide LSTM$^{\mathcal{S}}$ to make better predictions over the unified tags for the complete task.

The three key components are named **B**oundary **G**uidance (BG) component, **S**entiment **C**onsistency (SC) component and **O**pinion-**E**nhanced (OE) Target Word Detection component. The BG component takes the advantages of the boundary information provided by the auxiliary task to guide the LSTM$^{\mathcal{S}}$ for predicting the unified tags more accurately. The SC component is empowered with a gate mechanism to explicitly integrate the features of the previous word into the current prediction, aiming at maintaining the sentiment consistency within a multi-word opinion target. In order to provide boundary information of higher quality, the OE component, following the oberservation that "opinion targets and opinion words always co-occur", performs another auxiliary binary classification task to determine if the current word is a target word.

Target Boundary Guided TBSA We employ LSTM$^{\mathcal{S}}$ with softmax decoding layer for the prediction of the tag sequence. It is observed that the boundary tag can provide important clues for the unified tag prediction. For example, if the current boundary tag is B, denoting the beginning of an opinion target, then the corresponding unified tag can only be B-POS, B-NEG or B-NEU. Thus, we introduce an

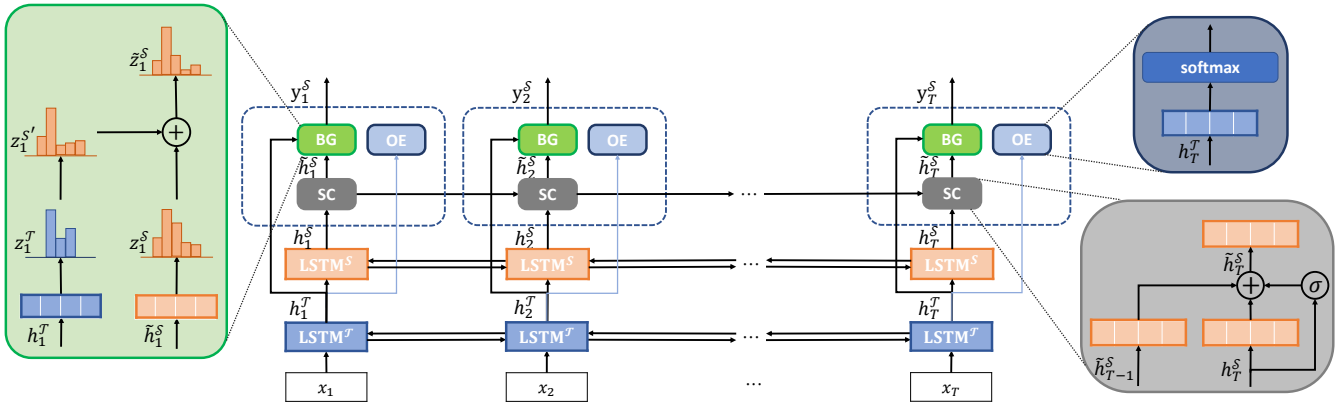

Figure 1: Architecture of the proposed framework.

additional network LSTM$^{\mathcal{T}}$ for the target boundary prediction, where the valid tag set $\mathcal{Y}^{\mathcal{T}}$ is {B, I, E, S, O}. We link these two LSTM layers so that the hidden representations generated by the LSTM$^{\mathcal{T}}$ can be directly fed to LSTM$^{\mathcal{S}}$ as guidance information. Specifically, their hidden representations $h_t^{\mathcal{T}} \in \mathbb{R}^{\dim_h^{\mathcal{T}}}$ and $h_t^{\mathcal{S}} \in \mathbb{R}^{\dim_h^{\mathcal{S}}}$ at the t-th time step ($t \in [1, T]$) are calculated as follows:

$$\begin{aligned} h_t^{\mathcal{T}} &= [\overrightarrow{\text{LSTM}}^{\mathcal{T}}(x_t); \overleftarrow{\text{LSTM}}^{\mathcal{T}}(x_t)], \\ h_t^{\mathcal{S}} &= [\overrightarrow{\text{LSTM}}^{\mathcal{S}}(h_t^{\mathcal{T}}); \overleftarrow{\text{LSTM}}^{\mathcal{S}}(h_t^{\mathcal{T}})], t \in [1, T]. \end{aligned} \tag{1}$$

The probability scores $z_t^{\mathcal{T}} \in \mathbb{R}^{|\mathcal{Y}^{\mathcal{T}}|}$ over the boundary tags are calculated by a fully-connected softmax layer:

$$z_t^{\mathcal{T}} = \mathbf{p}(y_t^{\mathcal{T}} | x_t) = \text{Softmax}(\mathbf{W}^{\mathcal{T}} h_t^{\mathcal{T}}). \tag{2}$$

where the Softmax denotes the softmax activation function and $\mathbf{W}^{\mathcal{T}}$ is the model parameter. Similarly, the scores over the unified tags $z_t^{\mathcal{S}} \in \mathbb{R}^{|\mathcal{Y}^{\mathcal{S}}|}$ are obtained as below:

$$z_t^{\mathcal{S}} = \mathbf{p}(y_t^{\mathcal{S}} | h_t^{\mathcal{T}}) = \text{Softmax}(\mathbf{W}^{\mathcal{S}} h_t^{\mathcal{S}}). \tag{3}$$

As mentioned above, the boundary information is supposed to be useful for improving the performance of LSTM$^{\mathcal{S}}$. (**?**) incorporated such boundary information by adding hard boundary constraints in the decoding step of the CRFs model. However, their prediction results are not promising. One reason is that their model employs a hard constraint which is prone to propagating the errors from the tagger of the boundary detection task and thus it decreases the performance of the TBSA tagger. Different from their way of imposing hard constraints, our proposed BG component can absorb the boundary information via boundary guided transition and automatically determine its proportions in the final tagging decision based on the confidence of the target boundary tagger. Firstly, the BG component encodes the constraints into a transition matrix $\mathbf{W}^{tr} \in \mathbb{R}^{|\mathcal{Y}^{\mathcal{T}}| \times |\mathcal{Y}^{\mathcal{S}}|}$. As we have no prior knowledge about the transition probabilities between the boundary tags and the unified tags, we initially set them equally as follows:

$$\mathbf{W}_{i,j}^{tr} = \begin{cases} \frac{1}{|\mathcal{B}_i|}, & \text{if } j \in \mathcal{B}_i \\ 0, & \text{Otherwise} \end{cases} \tag{4}$$

where $\mathcal{B}_i$ is the set of valid unified tags coherent with the boundary tag i. In this transition matrix, a non-zero element, e.g., $\mathbf{W}_{\text{B,B-POS}}^{tr}$, denotes the probabilities of the unified tags given the boundary tag, and a zero element, e.g., $\mathbf{W}_{\text{B,I-NEG}}^{tr}$, suggests that the unified tag cannot be inferred through this transition. After encoding the constraints, the next step is to guide the unified tag prediction with the boundary information. We directly propagate such information to the TBSA tagger by mapping the probability scores of the boundary tag $z_t^{\mathcal{T}}$ to the unified tag space. The transition-based sentiment score $z_t^{\mathcal{S}'} \in \mathbb{R}^{|\mathcal{Y}^{\mathcal{S}}|}$ is obtained as follows:

$$z_t^{\mathcal{S}'} = (\mathbf{W}^{tr})^{\top} z_t^{\mathcal{T}} \tag{5}$$

where the transition operation is equivalent to the linear combination of the row vectors in the transition matrix $\mathbf{W}^{tr}$. Assuming $z_t^{\mathcal{T}} = [1, 0, 0, 0, 0]$ (i.e., taking the tag B), the result of the transition is exactly the row vector $\mathbf{W}_{\text{B},:}^{tr}$. As the unified tag can be partially derived from the boundary tag, a natural question is how to determine the proportions of the transition-based unified tagging scores $z_t^{\mathcal{S}'}$. Intuitively, if the target boundary score $z_t^{\mathcal{T}}$ is nearly uniform, suggesting that the boundary tagger is not confident to its prediction, the obtained distribution over the unified tags, i.e., $z_t^{\mathcal{S}'}$, will also be close to a uniform distribution and has little meaningful information for the sentiment prediction. To avoid such uninformative boundary transitions, we calculate a proportion score $\alpha_t \in \mathbb{R}$ based on the confidence c_t of the target boundary tagger:

$$\begin{aligned} c_t &= (z_t^{\mathcal{T}})^{\top} z_t^{\mathcal{T}} \\ \alpha_t &= \epsilon c_t \end{aligned} \tag{6}$$

where the hyper-parameter ϵ denotes the maximum proportions that the boundary-based scores $z_t^{\mathcal{S}'}$ occupy in the tagging decision. Obviously, c_t will be down-weighted if the boundary scores are uniformly distributed. The maximum confidence value is reached if $z_t^{\mathcal{T}}$ is a one-hot vector. The final scores are obtained by combining the boundary-based

and model-based unified tagging scores:

$$z_t^{\mathcal{S}} = \alpha_t z_t^{\mathcal{S}'} + (1 - \alpha_t) z_t^{\mathcal{S}}. \quad (7)$$

Maintaining Sentiment Consistency In the traditional target sentiment classification task, the sentiments towards the different words in a given multi-word opinion target are assumed to be identical. However, in the complete TBSA task, such sentiment consistency is not guaranteed since the task is formulated as a sequence tagging/labeling problem. Taking the sentence in Table 1 as an example, there is still some possibility that the word "Processor" is labeled with an `E-NEG` tag due to the independent tagging decisions made by LSTMs. To maintain the sentiment consistency within the same opinion target, we propose to predict the current unified tag using both of the features from the current and the previous time steps. Specifically, we design a Sentiment Consistency (SC) component with a gate mechanism to combine these two feature vectors:

$$\begin{aligned} \tilde{h}_t^{\mathcal{S}} &= g_t \odot h_t^{\mathcal{S}} + (1 - g_t) \odot \tilde{h}_{t-1}^{\mathcal{S}} \\ g_t &= \sigma(\mathbf{W}^g h_t^{\mathcal{S}} + \mathbf{b}^g) \end{aligned} \quad (8)$$

where $\mathbf{W}^g$ and $\mathbf{b}^g$ are learnable parameters of the SC component, and $\odot$ denotes the element-wise multiplication. σ is the sigmoid function. Through the gating, the previous features are considered in the current predictions and such indirect bi-gram dependency can help reduce the probability that the words within the same target hold different sentiments.

Auxiliary Target Word Detection A good boundary tagger for opinion targets is crucial for producing the boundary information of high quality. Here, we introduce the **OE** component to learn a more robust boundary tagger from another view of the training data. As defined in (**?**; **?**; **?**), opinion targets are always collocated with opinion words. Inspired by this, we regard the word as a target word if there is at least one opinion word within the context window of fixed-size s of this word. Then, we train an auxiliary token-level classifier for discriminating target words and non-target words based on the distantly supervised labels and the boundary representations $h_t^{\mathcal{T}}$ are further refined with such supervision signals. The computational process of the **OE** component is below:

$$\begin{aligned} z_t^{\mathcal{O}} &= \text{Softmax}(\mathbf{W}^o h_t^{\mathcal{T}}) \\ y_t^{\mathcal{O}} &= \arg\max_y z_t^{\mathcal{O}} \end{aligned} \quad (9)$$

where $\mathbf{W}^o$ is the model parameter.

Model Training

All the components in our framework are differentiable, thus, the whole framework can be efficiently trained with gradient-based methods. Word/Token-level cross-entropy error is employed as the loss function:

$$\mathcal{L}^{\mathcal{I}} = -\frac{1}{T} \sum_{t=1}^{T} \mathbb{I}(y_t^{\mathcal{I},g}) \circ \log(z_t^{\mathcal{I}}) \quad (10)$$

where $\mathcal{I}$ is the symbol of task indicator and its possible values are $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{O}$. $\mathbb{I}(y)$ represents the one-hot vector with the y-th component being 1 and $y_t^{\mathcal{I},g}$ is the gold standard tag for the task $\mathcal{I}$ at the time step t. Then, the losses from the main TBSA task and the two auxiliary tasks are aggregated to form the training objective $\mathcal{J}(\theta)$ of the framework:

$$\mathcal{J}(\theta) = \mathcal{L}^{\mathcal{S}} + \mathcal{L}^{\mathcal{T}} + \mathcal{L}^{\mathcal{O}}. \quad (11)$$

Dataset		Train	Dev	Test	Total
$\mathbb{D}_{\text{L}}$	# `POS`	883	104	339	1326
	# `NEG`	754	106	130	990
	# `NEU`	404	46	165	615
$\mathbb{D}_{\text{R}}$	# `POS`	2337	270	1524	4131
	# `NEG`	942	93	500	1535
	# `NEU`	614	50	263	927
$\mathbb{D}_{\text{T}}$	# `POS`		-		692
	# `NEG`		-		263
	# `NEU`		-		2244

Table 2: Statistics of the datasets.

Experiments

Dataset

Our model is evaluated on two product review datasets from SemEval ABSA challenges (**?**; **?**; **?**) and the Twitter dataset. Table 2 gives the statistics of these benchmark datasets. $\mathbb{D}_{\text{L}}$ (SemEval 2014) contains reviews from the laptop domain and the train-test split is the same as the original dataset. $\mathbb{D}_{\text{R}}$ is the union set of the restaurant datasets from SemEval ABSA challenge 2014, 2015 and 2016. The new training dataset is obtained by merging the three years' training datasets and the new testing set is built in the same way. $\mathbb{D}_{\text{T}}$ consists of tweets collected by (**?**). The ground truth of the opinion target mentions and their sentiments are provided in these datasets. For $\mathbb{D}_{\text{L}}$ and $\mathbb{D}_{\text{R}}$, we regard 10% randomly held-out training data as the development set. For $\mathbb{D}_{\text{T}}$, we report the ten-fold cross validation results, as done in (**?**; **?**), since there is no standard train-test split for this dataset.

The gold standard boundary annotations are available for the auxiliary target boundary prediction task. For another auxiliary task, namely, opinion-based target word detection, we employ the existing opinion lexicon[1] to provide the opinion words.

The evaluation metric measures the standard precision (P), recall (R) and F1 score based on the **exact** match, which means that an output segment is considered to be correct only if it exactly matches with the gold standard span of the target mention and the corresponding sentiment.

Compared Models

We compare our framework with the following methods:

- **CRF**-{**pipeline**, **joint**, **unified**} (**?**): Conditional Random Fields (CRF) based sequence tagger[2]. "pipeline" denotes the pipeline approach. "joint" and "unified" are the models following the joint tagging scheme and unified tagging scheme respectively.

[1] http://mpqa.cs.pitt.edu/

[2] http://www.m-mitchell.com/code/index.html

	Model	$\mathbb{D}_{\text{L}}$			$\mathbb{D}_{\text{R}}$			$\mathbb{D}_{\text{T}}$		
		P	R	F1	P	R	F1	P	R	F1
Existing Baselines	CRF-joint	57.38	35.76	44.06	60.00	48.57	53.68	43.09	24.67	31.35
	CRF-unified	59.27	41.86	49.06	63.39	57.74	60.43	48.35	19.64	27.86
	NN-CRF-joint	55.64	34.48	45.49	61.56	50.00	55.18	44.62	35.84	39.67
	NN-CRF-unified	58.72	45.96	51.56	62.61	60.53	61.56	46.32	32.84	38.36
Pipeline Baselines	CRF-pipeline	59.69	47.54	52.93	52.28	51.01	51.64	42.97	25.21	31.73
	NN-CRF-pipeline	57.72	49.32	53.19	60.09	61.93	61.00	43.71	37.12	40.06
	HAST-TNet	56.42	54.20	55.29	62.18	73.49	67.36	46.30	49.13	47.66
Unified Baselines	LSTM-unified	57.91	46.21	51.40	62.80	63.49	63.14	51.45	37.62	43.41
	LSTM-CRF-1	58.61	50.47	54.24	66.10	66.30	66.20	51.67	44.08	47.52
	LSTM-CRF-2	58.66	51.26	54.71	61.56	67.26	64.29	53.74	42.21	47.26
	LM-LSTM-CRF	53.31	59.4	56.19	68.46	64.43	66.38	43.52	52.01	47.35
OURS	Base model	60.00	46.85	52.61	61.48	66.16	63.73	53.02	41.47	46.50
	Base model + **BG**	58.58	50.63	54.31	67.51	66.42	66.96	52.26	43.84	47.66
	Base model + **BG** + **SC**	58.95	53.00	55.81	63.95	69.65	66.68	53.12	43.60	47.79
	Base model + **BG** + **OE**	63.43	49.53	55.62	62.85	66.77	65.22	53.10	43.50	47.78
	Full model	61.27	54.89	**57.90**$^{\natural,\sharp}$	68.64	71.01	**69.80**$^{\natural,\sharp}$	53.08	43.56	**48.01**$^{\sharp}$

Table 3: Main results of the complete TBSA task. "Base model" refers to the stacked LSTMs. The markers $\natural$ and $\sharp$ refer to our full model significantly outperforms **HAST-TNet** and **LM-LSTM-CRF** respectively.

- **NN-CRF**-{**pipeline**, **joint**, **unified**} (**?**): Enhanced CRF models[3] armed with word embeddings and neural network feature extractors.
- **HAST-TNet**: HAST (**?**) and TNet (**?**) are the current state-of-the-art models on the tasks of target boundary detection and target sentiment classification respectively. HAST-TNet is the pipline approach of these two models. We use the officially released codes[4] to produce the results.
- **LSTM-unified**: the standard LSTM model adopting the unified tagging scheme.
- **LSTM-CRF-1** (**?**): LSTM model with CRF decoding layer and no feature engineering is needed. We run the officially released code [5] and utilize the unified tag set to reproduce the results.
- **LSTM-CRF-2** (**?**): LSTM-CRF-2 is similar to LSTM-CRF-1. The difference is that LSTM-CRF-2 employs CNN rather than LSTM to learn the character-level word representations. We run the released code[6] to reproduce the results.
- **LM-LSTM-CRF** (**?**): Language model enhanced LSTM-CRF model. It is a competitive model in several sequence tagging tasks. We rerun their code[7] and report the tagging results based on the unified tagging scheme.

[3]https://github.com/SUTDNLP/NNTargetedSentiment

[4]Available at: https://github.com/lixin4ever/HAST and https://github.com/lixin4ever/TNet respectively.

[5]https://github.com/glample/tagger

[6]https://github.com/XuezheMax/NeuroNLP2

[7]https://github.com/LiyuanLucasLiu/LM-LSTM-CRF

Experiment Settings

Word Embeddings We use `GloVe.840B.300d` [8] released by (**?**) to initialize the word embeddings, fine-tuned during training. The embeddings of the out-of-vocabulary words are sampled from the uniform distribution $\mathcal{U}$(-0.25, 0.25) (**?**).

Weight Initializations The weight matrices in the LSTM units are initialized by following the Glorot Uniform strategy (**?**) and the others are randomly sampled from the uniform distribution $\mathcal{U}$(-0.2, 0.2). Besides, all biases are initialized as 0's.

Optimization Our models are trained up to 50 epochs with Adam (**?**), with $\beta_1 = \beta_2 = 0.9$, and the initial learning rate $\eta_0 = 10^{-3}$. The decay rate is kept the same as the setting in (**?**). We apply dropout on word embeddings and the ultimate features for prediction. The dropout rates are empirically set as 0.5. The model obtaining the best F1 score on the development set is selected for producing the testing results.

Others Both of the dimension of the hidden representations $\text{dim}_h^{\mathcal{T}}$ and $\text{dim}_h^{\mathcal{S}}$ are 50. The maximum proportion ϵ of the boundary-based scores is 0.5. The size of the context window s in the opinion-based target word detection component is 3. The tuning details of ϵ and s are given later.

Results and Analysis

Main Results Table 3 presents our comparisons with other methods for the complete TBSA task. To make the comparison fair, we use `GloVe.840B.300d` as the pre-trained word embeddings for all the baselines requiring word embedding input on all of the datasets. Besides, we align the

[8]https://nlp.stanford.edu/projects/glove/

Input	Base model		Base model + **BG**		**Full model**	
	Target	Complete	Target	Complete	Target	Complete
1. And the fact that it comes with an [*i5 processor*]$_{\text{POS}}$ definitely speeds things up	*i5 processor*	[*processor*]$_{\text{POS}}$ (✗)	*i5 processor*	[*i5 processor*]$_{\text{POS}}$	*i5 processor*	[*i5 processor*]$_{\text{POS}}$
2. There were small problems with [*mac office*]$_{\text{NEG}}$.	*mac office*	[*mac*]$_{\text{NEG}}$ (✗)	*mac office*	[*mac office*]$_{\text{NEG}}$	*mac office*	[*mac office*]$_{\text{NEG}}$
3. The [*teas*]$_{\text{POS}}$ are great and all the [*sweets*]$_{\text{POS}}$ are homemade	*teas*, *sweets*	[*teas*]$_{\text{POS}}$, [*sweets*]$_{\text{POS}}$	*teas*, *sweets*, *homemade* (✗)	[*teas*]$_{\text{POS}}$, [*sweets*]$_{\text{POS}}$, [*homemade*]$_{\text{POS}}$ (✗)	*teas*, *sweets*	[*teas*]$_{\text{POS}}$, [*sweets*]$_{\text{POS}}$
4. I love the [*form factor*]$_{\text{POS}}$	NONE	NONE	NONE	NONE	*form factor*	[*form factor*]$_{\text{POS}}$
5. I blame the [*Mac OS*]$_{\text{NEG}}$.	*Mac OS*	[*Mac*$_{\text{NEG}}$ *OS*$_{\text{NEU}}$] (✗)	*Mac OS*	[*Mac*$_{\text{NEG}}$ *OS*$_{\text{POS}}$] (✗)	*Mac OS*	[*Mac OS*]$_{\text{NEG}}$
6. Also, I personally wasn't a fan of the [*portobello and asparagus mole*]$_{\text{NEG}}$.	*portobello and asparagus mole*	[*portobello*$_{\text{NEG}}$ *and*$_{\text{NEG}}$ *asparagus*$_{\text{NEG}}$ *mole*$_{\text{NEU}}$] (✗)	*portobello and asparagus mole*	[*portobello*$_{\text{NEG}}$ *and*$_{\text{NEG}}$ *asparagus*$_{\text{NEU}}$ *mole*$_{\text{NEU}}$] (✗)	*portobello and asparagus mole*	[*portobello and asparagus mole*]$_{\text{NEG}}$

Table 4: Case analysis. The "Target" column contains the results from the auxiliary task of target boundary detection. The "Complete" column presents the output of the complete TBSA task, but note that we only show the sentiment part of the unified labels (i.e., POS, NEG, and NEU) and use brackets to indicate the boundary. The marker ✗ denotes the incorrect prediction.

train/dev/test configurations for all methods. The experimental results suggest that our proposed framework consistently gives the best F1 score across all datasets and significantly outperforms the strongest baselines in most cases.

Compared to HAST-TNet, the pipeline of two state-of-the-art models, our proposed framework achieves 2.6%, 2.4% and 0.40% absolute gains on $\mathbb{D}_{\text{L}}$, $\mathbb{D}_{\text{R}}$ and $\mathbb{D}_{\text{T}}$ respectively, suggesting that a carefully-designed integrated model can be more effective than the pipeline approaches on the TBSA task. Three competitive unified sequence taggers (see the third block in Table 3) are also introduced into the comparative study. Again, our framework outperforms the best of them by 1.7%, 3.4% and 0.5% on the benchmark datasets. We notice that the improvement of our framework on the Twitter dataset is marginal in contrast with the unified baselines. The small gap is reasonable since these models employ additional component (e.g., LSTM or CNN) to learn the character-level word representations, whose capability for representing out-of-vocabulary words has been verified in (**?**; **?**), while our framework only utilizes the word-level features provided by the pre-trained word embeddings. Similar observation is captured in the comparison with HAST-TNet. We attribute this to the superior modeling power of the CNN applied in TNet when processing the ungrammatical sentences such as tweets and micro-blogs, as pointed out in (**?**).

We also notice that the performances of the CRF-based models, especially the recall (R) scores, are quite poor. Armed with the pre-trained word embeddings and neural network feature extractor, the models are slightly improved but the scores are still not promising.

Effectiveness of the Proposed Components To investigate the effectiveness of the designed components, we conduct ablation study on the proposed framework and the results are listed in the last block of the Table 3. Let us start the discussion from the base model, namely, the stacked LSTMs. We find that the base model always gives superior performance compared to the LSTM-unified. This result indicates that the boundary information predicted by the auxiliary LSTM indeed increases the F1 score of the complete TBSA task. With the help of the BG component, the performances are improved more significantly and the way we impose the boundary constraints proves effective for yielding more true positives. Another interesting finding is that introducing the component **SC** or **OE** individually into the "Base model + **BG**" does not bring in too much gains on F1 measure and even hurts the prediction performance on $\mathbb{D}_{\text{R}}$. But putting them together, i.e., the "Full model", leads to the new state-of-the-art result. This result illustrates the necessity of both of the **SC** and **OE** components in the boundary guided TBSA. Considering the "Base model + **BG** + **SC**", the quality of the boundary information may not be accurate without the clues from the **OE** component, and thus, the **SC** component tends to incorrectly align the sentiments of both the target words and non-target words. For the "Base model + **BG** + **OE**", the quality of the boundary information obtained from the LSTM$^{\mathcal{T}}$ is improved but the sentiments of the words within the same target are not fully consistent compared to the "Full model" armed with **SC** component. In summary, the **SC** component and the **OE** component are complementary to some extent when they are added into the boundary-guided "Base model + **BG**".

Case Analysis Table 4 gives some prediction examples of the base model (i.e., the stacked LSTMs) and the models empowered with our proposed components. As observed in the first input and the second input, the "Base model" correctly predicts the target boundary but it fails to produce the right target sentiments, suggesting that linking the two LSTMs for the target boundary prediction and the TBSA task is still insufficient for exploiting the boundary information to improve the performance of the complete TBSA. The "Base model+**BG**" and the "Full model", where the boundary constraints are properly imposed via our BG component, can correctly handle these two cases. Although the boundary information can guide the model to predict the sentiment more accurately, there is the possibility that only using the BG component (i.e., "Base model+**BG**") inherits the errors from the lower boundary detection task, e.g., the third and the fourth input. Thus, the boundary information of high

quality is crucial for improving the upper TBSA task and our OE component can serve as a simple but effective solution. Besides, we find that maintaining sentiment consistency within the same target mention, especially for whose with several words (e.g., "*portobello and asparagus mole*" in the last input), is difficult for the "Base model" and "Base model+**BG**", while our "Full model" alleviates this issue by employing the SC component to make predictions based on the features from the current and the previous time step.

Impact of ϵ and s Here, we investigate the impacts of the maximum proportion ϵ of the boundary-based scores and the window size s on the prediction performance. Specifically, the experiments are conducted on the development set of $\mathbb{D}_{\mathrm{R}}$, the largest benchmark dataset. We vary ϵ from 0.3 to 0.7, increased by 0.1, and two extreme values 0.0 and 1.0 are also included. The range of the window size s is 1 to 5. According to the results given in Figure 2, we observe that the best results are obtained at ϵ=0.5. The ϵ value basically affects the importance of the sentiment scores from the BG component in the final tagging decision and 0.5 is a good trade-off between absorbing boundary information and eliminating noises. We also observe that a moderate value of s (i.e., s = 3) is the best for the TBSA task, probably because too large s may enforce the model to attend the larger context and increase the possibility of associating with irrelevant opinion words, on the other hand, too small s is likely not sufficient to involve the potential opinion words.

Related Works

As mentioned in Introduction, Target-based Sentiment Analysis are usually divided into two sub-tasks, namely, the Opinion Target Extraction task (OTE) and the Target Sentiment Classification (TSC) task. Although these two subtasks are treated as separate tasks and solved individually in most cases, for more practical applications, they should be solved in one framework. Given an input sentence, the output of a method should contain not only the extracted opinion targets, but also the sentiment predictions towards them. Some previous works attempted to discover the relationship between these two sub-tasks and gave a more integrated solution for solving the complete TBSA task. Concretely, (**?**) employed Conditional Random Fields (CRF) together with hand-crafted linguistic features to detect the boundary of the target mention and predict the sentiment polarity. (**?**) further improved the performance of the CRF based method by introducing a fully connected layer to consolidate the linguistic features and word embeddings. However, they found that a pipeline method can beat both of the model with joint training and the unified model. In this paper, we reexamine the task, and proposed a new unified solution which outperforms all previous reported methods.

Conclusions

We investigate the complete task of Target-Based Sentiment Analysis (TBSA), which is formulated as a sequence tagging problem with a unified tagging scheme in this paper. The basic architecture of our framework involves two

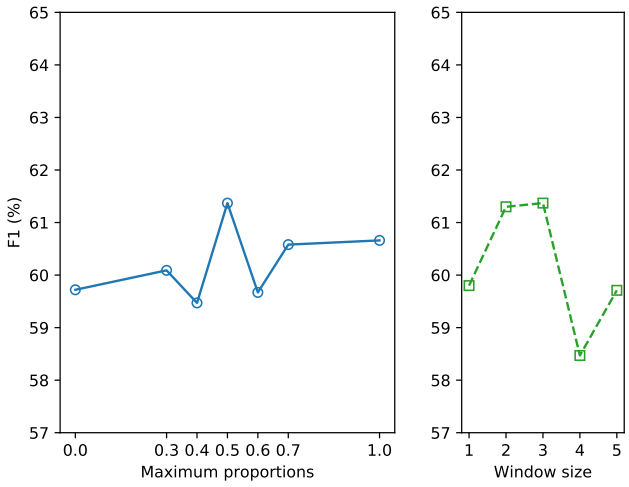

Figure 2: F1 scores (%) on the development set of $\mathbb{D}_{\mathrm{R}}$ with different ϵ and s values.

stacked LSTMs for performing the auxiliary target boundary detection and the complete TBSA task respectively. On top of the base model, we designed two components to take the advantage of the target boundary information from the auxiliary task and maintain the sentiment consistency of the words within the same target. To ensure the quality of the boundary information, we employ an auxiliary opinion-based target word detection component to refine the predicted target boundaries. Experimental results and case studies well illustrate the effectiveness of our proposed framework, and a new state-of-the-art result of this task is achieved. We publicly release our implementation at `https://github.com/lixin4ever/E2E-TBSA`.

Ab perferendis voluptatibus ea libero voluptas, architecto a repellendus hic.Dolor illum dicta soluta consequuntur nostrum eum odio mollitia ex officiis, voluptatum dignissimos ipsam voluptatem aperiam facere optio soluta porro a tempore unde?Facilis molestiae laboriosam ratione, consequatur sit repellendus perspiciatis, animi atque quisquam assumenda recusandae optio voluptates porro ad aliquam?Cupiditate voluptates error ipsa itaque labore, similique aspernatur cupiditate.Consequuntur modi atque quam numquam earum voluptate molestiae, magni dolores earum voluptate ratione consequatur consequuntur doloribus minima, velit dolorum beatae iusto aliquam esse id quibusdam soluta, nobis fuga iusto debitis vero dignissimos perspiciatis.Odit mollitia voluptatibus nemo nam dignissimos, possimus explicabo error officia odio illum optio enim beatae consequatur veniam, impedit soluta quod error excepturi nisi tenetur repudiandae at voluptatibus reiciendis, sit ullam accusantium optio, quisquam hic suscipit libero eligendi non voluptatem voluptatibus labore.Numquam nisi laborum nobis beatae provident, perspiciatis doloribus totam saepe nostrum quae quos praesentium id pariatur voluptatibus.Inventore itaque sunt quibusdam in fugiat incidunt soluta consequatur hic, enim vitae tempora quam at aliquam debitis provident, adipisci enim minus dolores blanditiis per-

spiciatis reprehenderit culpa quod recusandae eum, incidunt ad dignissimos laboriosam tempora pariatur reprehenderit necessitatibus enim sequi?