A Shared-Private Representation Model with Coarse-to-Fine Extraction for Target Sentiment Analysis

Peiqin Lin[1] **Meng Yang**[1,2*]
[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2]Key Laboratory of Machine Intelligence and Advanced Computing (SYSU), Ministry of Education
lpq29743@gmail.com, yangm6@mail.sysu.edu.cn

Abstract

Target sentiment analysis aims to detect opinion targets along with recognizing their sentiment polarities from a sentence. Some models with span-based labeling have achieved promising results in this task. However, the relation between the target extraction task and the target classification task has not been well exploited. Besides, the span-based target extraction algorithm has a poor performance on target phrases due to the maximum target length setting or length penalty factor. To address these problems, we propose a novel framework of Shared-Private Representation Model (SPRM) with a coarse-to-fine extraction algorithm. For jointly learning target extraction and classification, we design a Shared-Private Network, which encodes not only shared information for both tasks but also private information for each task. To avoid missing correct target phrases, we also propose a heuristic coarse-to-fine extraction algorithm that first gets the approximate interval of the targets by matching the nearest predicted start and end indexes and then extracts the targets by adopting an extending strategy. Experimental results show that our model achieves state-of-the-art performance.

1 Introduction

Target sentiment analysis aims to detect the opinion targets explicitly mentioned in the sentences, referred to as target extraction, and predict the sentiment polarities over the opinion targets, referred to as target classification. For example, in the sentence "I love Windows 7 which is a vast improvement over Vista.", the user mentions two opinion targets, namely, "Windows 7" and "Vista", and expresses positive sentiment over the first target, and negative sentiment over the second one.

Traditional methods formulated the jointly target extraction and classification task as a sequence labeling task. Under the scheme of sequence tagging, some prevalent models, including Conditional Random Field (CRF) (Mitchell et al., 2013; Zhang et al., 2015; Li and Lu, 2017), Gated Recurrent Unit (GRU) (Ma et al., 2018), Long Short-Term Memory (LSTM) (Li et al., 2019a), Convolutional Neural Network (CNN) (He et al., 2019) and Bidirectional Encoder Representations from Transformers (BERT) (Li et al., 2019b), are applied. Although these methods have achieved improved results, they suffer from the sentiment inconsistency problem of sequence tagging scheme.

To address it, some methods with span-based labeling, which can assure the sentiment consistency within a span, have been proposed (Zhou et al., 2019; Hu et al., 2019). (Zhou et al., 2019) proposed a span-based loss to predict whether the target within a span is correct. (Hu et al., 2019) proposed a span-based model, which first predict the boundary of the targets and then predict the sentiment polarities based on the corresponding features. Although deep learning methods, especially span-based methods, have achieved promising results, there are still some issues:

1) The relation between target extraction and target classification is not well exploited. Previous methods applied either a shared encoding module (Ma et al., 2018) or two private encoding modules (Luo et al., 2019; Hu et al., 2019) to learn features for target extraction and target classification, thus weakening the ability to represent the relation between the two tasks. As shown in Fig. 1, there exist shared and private information between target extraction and target classification. Specifically, the semantic and syntactic information are essential for both tasks, so they are shared information. On the other hand, as for the target extraction sub-task, some information like noun and pronoun informa-

*Corresponding Author: Meng Yang

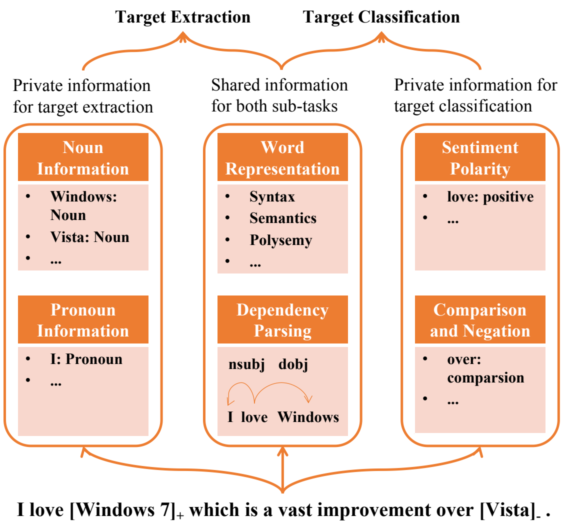

Figure 1: An example of shared and private information of target extraction and target classification

tion can be exploited but may interfere with target classification. Similarly, sentiment information may only be useful for target classification.

2) The span-based extraction algorithm still perform poorly on extracting target phrases. (Zhou et al., 2019) faces the trade-off problem between search space and target length. When it sets a small maximum target length, it may miss long target phrases. Conversely, setting a large maximum length will bring huge search space and many negative candidates. (Hu et al., 2019) adopts a heuristic algorithm a length penalty to avoid overlong targets. However, the length penalty make the model be incline to ignore target phrases.

To solve these issues, we propose a novel framework, namely Shared-Private Representation Model (SPRM) with a coarse-to-fine extraction algorithm. Inspired by (Bousmalis et al., 2016; Liu et al., 2016; Chen et al., 2018), we design a Shared-Private Network, which contains a shared encoding layer, namely Shared BERT (Devlin et al., 2018), and two private encoding layers, namely Target Extraction Long Short Term Memory (TE-LSTM) and Target Classification Long Short Term Memory (TC-LSTM). The two private networks can provide task-specific features and improve the ability of modeling the two sub-tasks. Moreover, we propose a coarse-to-fine extraction algorithm, which obtains the approximate intervals of targets by matching predicted start/end boundaries and then applies an extending strategy instead of a penalty factor for extracting target phrases correctly. The experiments on three benchmark datasets show that our model achieves state-of-the-art performance. Our contributions are summarized as follows:

- A Shared-Private Network is designed to learn the shared and private representations for both of the two sub-tasks;
- A coarse-to-fine extraction algorithm is proposed for target extraction to better extract target phrases;
- Experimental results show our model achieves start-of-the-art performance.

2 Related Work

(Mitchell et al., 2013) formulated the task of target sentiment analysis as a sequence tagging problem and proposed to use Conditional Random Field (CRF) with hand-crafted linguistic features. In the proposed method, three ways are designed to solve the problem, namely, pipeline way, collapsed way, and joint way. The pipeline way uses two independent models to extract targets and predict the sentiment of the extracted targets separately. As for the joint way, there are shared modules between the two sub-tasks that are jointly trained. Finally, the collapsed model combines the label of target extraction and target classification into the same label space, and predicts the collapsed label.

Based on (Mitchell et al., 2013), rule-based methods (Zhang et al., 2015; Li and Lu, 2017) and deep-learning-based methods (Ma et al., 2018; Li et al., 2019a; Luo et al., 2019; He et al., 2019) have been proposed to solve target sentiment analysis task with the sequence tagging scheme. Although these methods have achieved improved results, they suffer from the problem of huge search space and sentiment inconsistency of sequence tagging scheme (Hu et al., 2019).

To address it, some span-based models were proposed (Zhou et al., 2019; Hu et al., 2019), which solved the target sentiment analysis task by predicting the span of the targets. (Zhou et al., 2019) proposed a span-based loss to predict whether the target candidate with a span is a correct target. (Hu et al., 2019) proposed an extract-then-classify framework, which first extracts targets using a heuristic multi-span decoding algorithm and then classifies their polarities with corresponding summarized span representations. Compared to (Zhou et al., 2019), the extraction method proposed by (Hu et al., 2019) has solved the problem of huge

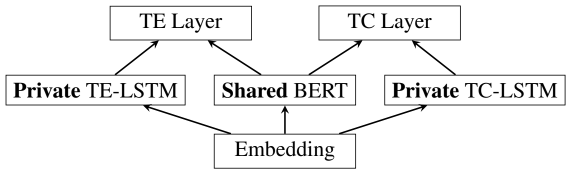

Figure 2: The overall architecture of SPRM. "TE" and "TC" denote "Target Extraction" and "Target Classification", respectively.

search space better and achieve better results. However, there are still some issues with it. For instance, (Hu et al., 2019) simply implements the joint model by employing a shared backbone for the two sub-tasks, which ignores the private information of each task. In addition, the heuristic multi-span decoding algorithm involves manually-setting thresholds for different datasets, and a length penalty factor for avoiding overlong targets, which is not suitable for extracting target phrases.

3 Model

To solve the aforementioned issues, we simultaneously learn shared and private features for the target extraction and classification in a unified framework, in which a coarse-to-fine extraction algorithm is designed. In this paper, we propose a novel model of Shared-Private Representation Model (SPRM) shown in Fig. 2, which encodes the shared and private information of the target extraction sub-task and the target classification sub-task effectively at a lower cost. Specifically, a Shared BERT Network is designed to encode as much shared information of both sub-tasks as possible, and two Private BiLSTMs are introduced to get the supplementary private representations for each task with fewer parameters than BERT. Moreover, we design a coarse-to-fine algorithm that first gets the approximate interval of the targets by matching the nearest predicted start and end indexes without any thresholds and then gets the final targets by extending the interval if the adjacent words are predicted as start/end boundaries. With the algorithm, targets can be extracted with reasonable length, since the nearest strategy avoids overlong targets while the extending strategy avoids missing target phrases.

3.1 Shared-Private Model

The overall architecture of Shared-Private Model is shown in Fig 2, which is composed by six components: an embedding layer, two Private BiLSTM networks for target extraction and target sentiment classification, a Shared BERT Network for both two sub-tasks, and the final layers of target extraction and target classification.

Given the sentence input, the embedding layer process it with the tokenization process and word-piece embeddings of BERT (Devlin et al., 2018), and obtain the input embeddings $\boldsymbol{E} \in R^{n \times d_e}$, where n is the length of the processed sequence and d_e is the size of embedding vectors.

For target sentiment analysis, both shared information of both sub-tasks and private information of each sub-tasks should be considered. Therefore, a shared network is designed to encode shared information between the two sub-tasks, such as semantic and syntactic information of the input sentence.

$$\boldsymbol{V}_s = f(\boldsymbol{E}) \tag{1}$$

where $f(\cdot)$ is the function of learning shared features and $\boldsymbol{V}_s$ is the learned feature.

At the same time, the task-specific private information of target extraction (e.g., whether a word is a noun) and target classification (e.g., sentiment information of each word) should be learned in private modules.

$$\boldsymbol{V}_{te} = g_{te}(\boldsymbol{E}), \boldsymbol{V}_{tc} = g_{tc}(\boldsymbol{E}) \tag{2}$$

where $g_{te}(\cdot)$ and $g_{tc}(\cdot)$ are the functions of learning private features of the target extraction task and the target classification task, $\boldsymbol{V}_{te}$ and $\boldsymbol{V}_{tc}$ are the private features.

Based on the shared and private features, fusion modules are designed to obtain the final features for the two sub-tasks.

$$\tilde{\boldsymbol{V}}_{te} = h_{te}(\boldsymbol{V}_s, \boldsymbol{V}_{te}), \tilde{\boldsymbol{V}}_{tc} = h_{tc}(\boldsymbol{V}_s, \boldsymbol{V}_{tc}) \tag{3}$$

where $h_{te}(\cdot)$ and $h_{tc}(\cdot)$ are the functions of fusing shared and private features of the target extraction task and the target classification task, $\tilde{\boldsymbol{V}}_{te}$ and $\tilde{\boldsymbol{V}}_{tc}$ are the final features, which are fed into output layers.

Finally, $\tilde{\boldsymbol{V}}_{te}$ and $\tilde{\boldsymbol{V}}_{tc}$ are fed into the Target Extraction Layer (TE-Layer) and Target Classification Layer (TC-Layer) to generate the predictions, respectively. The model is finally trained by minimizing the sum of the target extraction loss and polarity classification loss:

$$l = l_{TE} + l_{TC} \tag{4}$$

where l_{TE} and l_{TC} are the losses of the target extraction task and target classification task. Here, we omit an exhaustive description of the TC-Layer as it's same as the classification layer applied in (Hu et al., 2019), and readers can get more details from (Hu et al., 2019).

In the following subsections, we will detail the design of the aforementioned components, such as the shared module, the two private modules, the combination of shared and private modules, and the TE-Layer.

3.1.1 Shared BERT

As shared features are used in both target extraction and target classification, the shared module needs to have a strong ability of learning a shared representation. In addition, shared features generally portray common information between the two sub-tasks, like semantic and syntactic information, which also exist in other NLP tasks. Therefore, the prevalent model of Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), which is a pre-trained bidirectional Transformer encoder that achieves state-of-the-art performances across a variety of NLP tasks, is chosen as the shared network.

Given the embeddings $\boldsymbol{E}$, a series of stacked Transformer blocks are applied to project the input embeddings into a sequence of contextual vectors $\boldsymbol{V}_s \in R^{n \times d_s}$, where d_s is the dimension of outputs.

3.1.2 Private BiLSTM

Although the Shared BERT has captured powerful features for the two sub-tasks, these shared features are task-invariant but not task-specific. Therefore, private modules should be designed to learn private features for the two sub-tasks, respectively.

Since the Shared BERT has extracted as sufficient syntactic and semantic information as possible with a huge amount of parameters, we adopt Bidirectional Long Short Term Memory (BiLSTM), which captures the relationship between words in a sentence with fewer parameters than BERT, as the private modules. Specifically, we adopt two Private BiLSTM networks, namely TE-LSTM and TC-LSTM, to learn the private features for the tasks of target extraction and target sentiment classification, respectively. Taking the same embeddings $\boldsymbol{E}$ as inputs, we can obtain the output of BiLSTMs $\boldsymbol{V}_{te} \in R^{n \times 2d_p}$ and $\boldsymbol{V}_{tc} \in R^{n \times 2d_p}$, where d_p is the hidden size of the BiLSTM networks.

3.1.3 Combination of Shared and Private Features

Since the dimension of the Private BiLSTM output is twice than that of the shared BERT, we first project the outputs of shared and private modules into the same vector space by employing fully connected layers after the private modules:

$$\boldsymbol{V}'_{te} = FC_{te}(\boldsymbol{V}_{te}); \boldsymbol{V}'_{tc} = FC_{tc}(\boldsymbol{V}_{tc}) \qquad (5)$$

where $\boldsymbol{V}'_{te}, \boldsymbol{V}'_{tc} \in R^{n \times d_s}$. Then we simply apply concatenation operation to get the final features at a low cost.

$$\tilde{\boldsymbol{V}}_{te} = (\boldsymbol{V}_s; \boldsymbol{V}'_{te}); \tilde{\boldsymbol{V}}_{tc} = (\boldsymbol{V}_s; \boldsymbol{V}'_{tc}) \qquad (6)$$

3.2 Coarse-to-Fine Extraction Algorithm

(Hu et al., 2019) has proposed a heuristic algorithm based on the span-based labeling scheme and verified that the span-based labeling scheme performs better on target extraction compared to sequence tagging methods. However, the heuristic algorithm requires a manually-setting threshold for extracting targets and also has poor performance on target phrases due to the length penalty factor, which is designed to avoid overlong targets.

To address these issues, we propose a coarse-to-fine extraction algorithm. In the coarse-to-fine algorithm, the approximate interval of a target can be obtained by matching the nearest predicted start and end indexes rather than manually setting a threshold, and then the final target is extracted with a reasonable length by adopting an extending strategy, which extends the intervals if the adjacent words are predicted as start/end boundaries.

The implementation of the coarse-to-fine extraction algorithm is described in detail in the following subsections, and Table 1 shows how the algorithm is used in a concrete example. The coarse-to-fine extraction algorithm consist of three steps:

- **Boundary prediction** gets the predictions of start and end positions (Sec. 3.2.1);
- **Coarse extraction** generates approximate intervals of target candidates by the nearest strategy based on the prediction results (Sec. 3.2.2);
- **Fine extraction** generates the final targets with an extending strategy based on the approximate intervals of candidates (Sec. 3.2.3).

Input	Example	... easy to **integrate bluetooth devices**, and **USB devices** are recognized ...													
	ID	...	0	1	2	3	4	5	6	7	8	9	10	11	...
	Tokens	...	easy	to	integrate	blue	##tooth	devices	,	and	usb	devices	are	recognized	...
Boundary Prediction	$\boldsymbol{p}^s$	-	0.0005	-	0.8040	0.8515	-	-	-	-	0.9875	-	-	-	-
	$\boldsymbol{p}^e$	-	-	-	0.0060	-	-	0.9494	-	-	0.0171	0.8899	-	-	-
	$\boldsymbol{label}^s$	-	0	0	1	1	0	0	0	0	1	0	0	0	0
	$\boldsymbol{label}^e$	-	0	0	0	0	0	1	0	0	0	1	0	0	0
Coarse Extraction	Boundary Number	nb_s = 3 (3 true labels: {2, 3, 8}), nb_e = 2 (2 true labels: {5, 9}) $nb = max(nb_s, nb_e)$ = 3													
	Top nb Start/End Boundaries	$\boldsymbol{S}$={2, 3, 8} $\boldsymbol{E}$={5, 8, 9}													
	Target Candidates	$\boldsymbol{C}_s$ = {(2, 5), (3, 5), (8, 8)}, $\boldsymbol{C}_e$ = {(3, 5), (8, 8), (8, 9)} $\boldsymbol{C}$ = {(2, 5), (3, 5), (8, 8), (8, 9)}													
Fine Extraction	Extending Strategy	$\boldsymbol{C'}$ = {(2, 5), (8, 9)}													
	Target Number	nt_s = 2 (2 intervals: [2, 3] and [8, 8]), nt_e = 2 (2 intervals: [5, 5] and [9, 9]) $nt = round((nt_s + nt_e)/2)$ = 2													
	Top nt Targets	$\boldsymbol{O}$ = {(2, 5), (8, 9)}													
Output	Targets	["integrate bluetooth devices", "usb devices"]													

Table 1: An example for coarse-to-fine extraction algorithm. The input words are represented as their ids.

3.2.1 Boundary Prediction

As we have mentioned in Sec. 3.1, $\tilde{\boldsymbol{V}}_{te}$ is fed into the TE-Layer to generate the predictions, and then the loss of the target extraction task l_{TE} is computed. Here, the TE-Layer will be described in detail.

The start and end scores for each word in the sequence can be obtained by first applying fully connected layers and then using a sigmoid function:

$$\boldsymbol{g}^s = FC_s(\tilde{\boldsymbol{V}}_{te}),\ \boldsymbol{p}^s = sigmoid(\boldsymbol{g}^s) \quad (7)$$

$$\boldsymbol{g}^e = FC_e(\tilde{\boldsymbol{V}}_{te}),\ \boldsymbol{p}^e = sigmoid(\boldsymbol{g}^e) \quad (8)$$

Different to (Hu et al., 2019), we employ a sigmoid function instead of the softmax function to get the scores, because the sigmoid function is more suitable for binary classification, like predicting whether a word is a start/end here. Given the probabilities of start and end positions of each word, the corresponding labels denoting whether a word is the start/end boundary of a target can be computed by the following steps.

$$\boldsymbol{label}^s = \begin{cases} 1 & \text{if } \boldsymbol{p}^s \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$
$$\boldsymbol{label}^e = \begin{cases} 1 & \text{if } \boldsymbol{p}^e \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $\boldsymbol{p}^s = \{p^s_1, p^s_2, \ldots, p^s_n\}$ and $\boldsymbol{p}^e = \{p^e_1, p^e_2, \ldots, p^e_n\}$ are the start and end scores, respectively. Taking these two scores, the start labels $\boldsymbol{y}^s = \{y^s_1, y^s_2, \ldots, y^s_n\}$ and the end labels $\boldsymbol{y}^e = \{y^e_1, y^e_2, \ldots, y^e_n\}$ as inputs, we get the loss of target extraction:

$$l_{TE} = \sum_{i}^{n} (logloss(p^s_i, y^s_i) + logloss(p^e_i, y^e_i)) \quad (10)$$

where $logloss(p_i, y_i)$ is an error function defined as follows:

$$logloss(p_i, y_i) = \begin{cases} -log(p_i) & \text{if } y_i = 1 \\ -log(1 - p_i) & \text{if } y_i = 0 \end{cases} \quad (11)$$

3.2.2 Coarse Extraction

The coarse extraction step first gets top start/end boundaries and then generates the original set of target candidates by the nearest strategy, which matches the nearest predicted start and end boundaries without any thresholds.

Given the predicted labels of start and end positions, we can get the numbers of tokens predicted as start/end boundaries, namely nb_s and nb_e. Since enough candidates should be extracted to avoid missing correct candidates, we employ maximum function to compute the number of the boundaries nb which should be considered.

$$nb = max(nb_s, nb_e) \quad (12)$$

Therefore, the top nb candidates of start/end boundary from $\boldsymbol{p}^s$ and $\boldsymbol{p}^e$ are obtained and then the set of start/end candidates, namely $\boldsymbol{S}$ and $\boldsymbol{E}$, are generated.

Since a target generally consists of a few tokens, we apply nearest strategy to avoid overlong targets.

Using the nearest strategy, we match the nearest end index in $\boldsymbol{E}$ with each start boundary candidate to get the start target candidate set $\boldsymbol{C}_s$. Similarly, the end target candidate set $\boldsymbol{C}_e$ is also obtained. Finally, the approximate intervals of target candidates are obtained.

$$\boldsymbol{C} = \boldsymbol{C}_s \cup \boldsymbol{C}_e \tag{13}$$

3.2.3 Fine Extraction

To get the final targets, the fine extraction step first adopts an extending strategy and then selects targets based on start/end probabilities and the computed target number.

For target phrases, the boundaries of the nouns in them are usually predicted as start/end positions, too. For example, the token 'blue' of the target phrase 'integrate bluetooth devices' is predicted as the start position of a target, as shown in Table. 1. Therefore, an extending strategy shown in Algorithm 1 is designed to extract complete targets. In the extending strategy, every possible candidate can be extended on both the left side (line 3-4) and the right side (line 5-6) if the adjacent word is predicted as the start or end boundary.

Algorithm 1 Extending Strategy

```
Input: $\boldsymbol{C}$: the candidate set; $\boldsymbol{S}$: the start candidate set; $\boldsymbol{E}$: the end candidate set
Output: $\boldsymbol{C}'$: the extended candidate set
1: $\boldsymbol{C}' = \{\}, \boldsymbol{O} = \{\}$
2: for $(s_i, e_i)$ in $\boldsymbol{C}$ do
3:     while $s_i - 1 \geq 0$ and $s_i - 1 \in \boldsymbol{S}$ do
4:         $s_i = s_i - 1$
5:     while $e_i - 1 < n$ and $e_i + 1 \in \boldsymbol{E}$ do
6:         $e_i = e_i + 1$
7:     $\boldsymbol{C}' = \boldsymbol{C}' \cup (s_i, e_i)$
```

As is mentioned before, the boundaries of the nouns in target phrases are usually predicted as start/end position of a target. Therefore, we can observe that the model may predict one or a few start/end positions for a target, which are generally adjacent to each other. In other words, the numbers of intervals which contain only labels predicted as true start/end boundaries can be used to infer the number of extracted targets nt. Specifically, the interval numbers of $\boldsymbol{label}^s$ and $\boldsymbol{label}^e$, namely nt_s, nt_e, are computed first, and then we use the average value to estimate the number of the targets nt.

$$nt = round((nt_s + nt_e)/2) \tag{14}$$

Dataset		#+	#-	#0	Total
LAPTOP	Train	987	860	455	2302
	Test	339	130	165	634
REST	Train	2,610	1,037	667	4314
	Test	1,524	501	264	2289
TWITTER	-	703	274	2,266	3,243

Table 2: Dataset statistics. '+', '-', and '0' refer to the positive, negative, and neutral sentiment classes, respectively.

With the target number nt, we sort the extended candidate set $\boldsymbol{C}'$ in descending order with the addition of start and end probabilities and then choose the top nt candidates. Note that the candidates overlapped by the extracted targets will be removed while being chosen.

4 Experiments

4.1 Setup

4.1.1 Datasets

We conduct experiments on three benchmark datasets, as shown in Table. 2. LAPTOP contains product reviews from the laptop domain in SemEval 2014 (Pontiki et al., 2014). REST is the union set of the restaurant domain from SemEval 2014, 2015 and 2016 (Pontiki et al., 2015, 2016). TWITTER is built by (Mitchell et al., 2013), consisting of twitter posts. Following (Zhang et al., 2015; Li et al., 2019a; Hu et al., 2019), we report the ten-fold cross-validation results for TWITTER, as there is no train-test split. For each dataset, the gold target span boundaries are available, and the targets are labeled with sentiment polarities, namely positive (+), negative (-), and neutral (0).

4.1.2 Metrics

We adopt the precision (P), recall (R), and F1 score as evaluation metrics. A predicted target is correct only if it exactly matches the gold targets and the corresponding polarity. To separately analyze the performance of two sub-tasks, precision, recall, and F1 are also used for the target extraction subtask, while the accuracy (ACC) metric is applied to polarity classification.

4.1.3 Model Settings

We use the publicly available BERT-Base model as the shared BERT, and refer readers to (Devlin et al., 2018) for details on model sizes. The dimension sizes d_e, d_p and d_s are all 768. In addition, we use Adam optimizer (Kingma and Ba, 2014) with

	Model	LAPTOP			REST			TWITTER		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Sequence-tagging-based Method	CRF-pipeline	59.69	47.54	52.93	52.28	51.01	51.64	42.97	25.21	31.73
	CRF-joint	57.38	35.76	44.06	60.00	48.57	53.68	43.09	24.67	31.35
	CRF-collapsed	59.27	41.86	49.06	63.39	57.74	60.43	48.35	19.64	27.86
	NN-CRF-pipeline	57.72	49.32	53.19	60.09	61.93	61.00	43.71	37.12	40.06
	NN-CRF-joint	55.64	34.48	45.49	61.56	50.00	55.18	44.62	35.84	39.67
	NN-CRF-collapsed	58.72	45.96	51.56	62.61	60.53	61.56	46.32	32.84	38.36
	TAG-pipeline	65.84	67.19	66.51	71.66	76.45	73.98	54.24	54.37	54.26
	TAG-joint	65.43	66.56	65.99	71.47	75.62	73.49	54.18	54.29	54.20
	TAG-collapsed	63.71	66.83	65.23	71.05	75.84	73.35	54.05	54.25	54.12
	UNIFIED	61.27	54.89	57.90	68.64	71.01	69.80	53.08	43.56	48.01
	DOER	-	-	60.35	-	-	72.78	-	-	51.37
Span-based Method	Zhou SPAN	61.40	58.20	59.76	76.20	68.20	71.98	54.84	48.44	51.44
	Hu SPAN-pipeline	**69.46**	66.72	68.06	76.14	73.74	74.92	**60.72**	55.02	57.69
	Hu SPAN-joint	67.41	61.99	64.59	72.32	72.61	72.47	57.03	52.69	54.55
	Hu SPAN-collapsed	50.08	47.32	48.66	63.63	53.04	57.85	51.89	45.05	48.11
Our Model	**SPRM**	68.66	**68.77**	**68.72**	**77.78**	**80.60**	**79.17**	60.25	**58.76**	**59.45**

Table 3: Main results (%) on three benchmark datasets. State-of-the-art results are marked in bold.

	LAPTOP	REST
SPRM w/o Shared BERT	53.25	70.03
SPRM w/o Private BiLSTMs	66.72	78.78
SPRM w/o Aspect Extraction LSTM	66.20	78.74
SPRM w/o Aspect Classification LSTM	67.92	78.41
SPRM	**68.72**	**79.17**

Table 4: F1 results (%) on examining the effectiveness of Shared-Private Network.

a learning rate of 3e-5 and warmup over the first 10% steps. The batch size is 32 and a dropout probability of 0.1 is used.

4.1.4 Baselines

We compare SPRM with both sequence-tagging-based methods and span-based methods. The sequence-tagging-based methods includes CRF-{pipeline, joint, collapsed} (Mitchell et al., 2013), NN-CRF-{pipeline, joint, collapsed} (Zhang et al., 2015), TAG-{pipeline, joint, collapsed} (Hu et al., 2019), UNIFIED (Li et al., 2019a), DOER (Luo et al., 2019). The span-based methods are Zhou SPAN (Zhou et al., 2019) and Hu SPAN-{pipeline, joint, collapsed} (Hu et al., 2019).

4.2 Main Results

We report the results of SPRM and the baselines in Table. 3. Two main observations can be obtained from the table. Firstly, compared to SPAN-joint, SPRM improves the performance significantly by 4.13%, 6.70% and 4.90% on three datasets, since SPAN-joint ignores the private encoding components for the two sub-tasks and only apply a Shared BERT network. It shows that some private information for the two sub-tasks can be well obtained by applying two private encoding components. Secondly, SPRM achieves 0.66%, 4.25%, and 1.76% absolute gains on three datasets compared to the best SPAN method SPAN-pipeline, indicating the efficacy of the Shared BERT. Therefore, SPRM can get better performance with fewer parameters compared to SPAN-pipeline, which employs two separate BERT encoding network for target extraction and target classification, respectively.

	N_{para}	LAPTOP	REST
SPAN-pipeline + BERT-Large	671M	68.06	74.92
SPAN-joint + BERT-Large	336M	64.59	72.47
SPAN-joint + BERT-Base	109M	59.88	68.95
SPRM + BERT-Large	342M	**69.11**	79.08
SPRM + BERT-Base	131M	68.72	**79.17**

Table 5: F1 results (%) on LAPTOP and REST w.r.t different BERT backbone models.

	LAPTOP	REST
SPRM with CRF	59.55	75.34
SPRM with (Hu et al., 2019)	66.35	78.49
SPRM	**68.72**	**79.17**

Table 6: F1 results (%) on examining the effectiveness of Coarse-to-Fine Extraction Algorithm.

4.3 Effectiveness of Shared-Private Network

To verify the effectiveness of the Shared-Private Network, we conduct extensive experiments on the LAPTOP and REST datasets, and the experimental results is shown in Table. 4.

From the results, we observe that removing Shared BERT makes the performance worse since

BERT has a strong ability of learning powerful features. Although the model can perform well while just applying BERT, the Private BiLSTMs can also learn useful features for each sub-task to improve the performance. Specifically, the Private AE-LSTM is more effective than the Private AC-LSTM, as the performance of the former LSTM has a bigger decrease in performance.

Moreover, we plot the performance of SPAN and SPRM with respect to different BERT backbone networks in Table. 5 to further examine the effectiveness of the Shared-Private Network. We can observe that SPRM with BERT-Base achieves comparable results compared to SPRM with BERT-Large, while the performance of SPAN-joint with BERT-Base is significantly worse than that of SPAN-joint with BERT-Large. It shows that the introduction of private layers improves the performance with fewer parameters compared to using BERT-Large as the backbone network instead of BERT-Base. Besides, SPRM with BERT-Base outperforms SPAN-pipeline with BERT-Large, which uses almost 5 times the trainable parameters of SPRM with BERT-Base. Therefore, the introduction of Shared BERT can not only connect the task of target extraction and target classification to some extent but also reduce the parameter number.

4.4 Effectiveness of Coarse-to-Fine Extraction Algorithm

To verify the effectiveness of the coarse-to-fine extraction algorithm, we employ CRF and the heuristic algorithm proposed by (Hu et al., 2019) instead of our coarse-to-fine extraction algorithm on the LAPTOP and REST datasets, and the experimental results are shown in Table. 6.

Among the three extraction methods, CRF preforms worse since it suffers from the problems of huge search space. In addition, the coarse-to-fine extraction algorithm outperforms the heuristic extraction method of (Hu et al., 2019) as our model applies a flexible way to extract targets.

4.5 Analysis on Both Sub-Tasks

To analyze the performance of our model on target extraction and target sentiment classification, we compare our model with previous approaches designed for both of the two tasks and some state-of-the-art methods proposed for one of the sub-tasks, namely, DE-CNN (Xu et al., 2018) for target extraction and DMMN-SDCM (Lin et al., 2019) for target classification. The experimental results of target extraction and target classification are shown in Table. 7 and Table. 8, respectively.

Dataset	LAPTOP	REST	TWITTER
DE-CNN	81.59	-	-
TAG	**85.20**	84.48	73.47
SPAN	83.35	82.38	**75.28**
SPM	84.72	**86.71**	69.85

Table 7: F1 comparison of different approaches for target extraction.

Dataset	LAPTOP	REST	TWITTER
DMMN-SDCM	77.59	-	-
TAG	71.42	81.80	59.76
SPAN	81.39	89.95	75.16
SPM	**81.50**	**90.35**	**78.34**

Table 8: Accuracy comparison of different approaches for polarity classification.

On the task of target extraction, our model doesn't have the best performance on all of the three datasets. SPM outperforms SPAN by 1.37% and 4.33% on the LAPTOP and REST datasets, but has worse performance on the TWITTER dataset. And on the task of target sentiment classification, our model outperforms all the baselines by 0.11%, 0.40%, and 3.18% on three datasets. The experimental results show that one of the disadvantages of the joint model over the pipeline model is that it can make sure to perform best on the task of target sentiment analysis but can't perform best on both sub-tasks at the same time for guarantee.

4.6 Qualitative Analysis

Table. 9 shows some qualitative cases sampled from SPAN-pipeline and SPRM. We can observe that our model SPRM with the coarse-to-fine extraction algorithm can extract more accurate targets. The heuristic coarse-to-fine extraction algorithm computes the number of targets by the predict scores of start and end boundaries instead of a manually set threshold, so our method can be more precise with the number of targets. Take the example 6 in the table as an example, the correct targets, "Windows XP" and "Windows 7", are not extracted by SPAN-pipeline as the threshold filters them incorrectly, while our method extracts all the three correct targets as we infer the number of targets correctly. Example 1 is also a good example to confirm this. In addition, our algorithm adopts the extending strategy instead of the strategy of length

Examples	SPAN-pipeline	SPRM
1. All in all, the **[food]**$_+$ was great (except for the **[desserts]**$_-$).	[food]$_+$ (✓), None (✗)	[food]$_+$ (✓), [desserts]$_-$ (✓)
2. **[Vanison]**$_0$ was good but not amazing.	[Vanison]$_0$ (✓)	[Vanison]$_0$ (✓)
3. The **[selection of food]**$_+$ is excellent (I'm not used to having much choice at restaurants), and the **[atmosphere]**$_+$ is great.	[selection]$_+$ (✓), [food]$_+$ (✗), [atmosphere]$_+$ (✓)	[selection of food]$_+$ (✓), [atmosphere]$_+$ (✓)
4. Beware of the **[chili signed food items]**$_-$ not unless you want to call the fire department to douse the flames in your mouth.	[chili]$_-$ (✗), [food items]$_-$ (✗)	[chili signed food items]$_-$ (✓)
5. This mac does come with an **[extender cable]**$_0$ and I'm using mine right now hoping the **[cable]**$_+$ will stay nice for the many years I plan on using this mac.	[extender cable]$_0$ (✓), None (✗)	[extender cable]$_0$ (✓) [cable]$_+$ (✓)
6. I used **[Windows XP]**$_0$, **[Windows Vista]**$_0$, and **[Windows 7]**$_0$ extensively.	None (✗), [Windows Vista]$_0$ (✓), None (✗)	[Windows XP]$_0$ (✓), [Windows Vista]$_0$ (✓), [Windows 7]$_0$ (✓)
7. The only thing I miss is that my old Alienware laptop had **[backlit keys]**$_-$.	[backlit]$_-$ (✗), [keys]$_-$ (✗)	[backlit keys]$_-$ (✓)

Table 9: Case study. The extracted targets are wrapped in brackets with the predicted polarities given as subscripts. Correct and incorrect predictions are marked with ✓and ✗, respectively.

penalty, and it can avoid missing the targets which consist of a few words. Take the example 4 in the table as an example, the correct extracted target should be "chili signed food items", but SPAN-pipeline split the gold target entity to two separate targets because of its length penalty. However, our algorithm can extract the target "chili signed food items" correctly since we get the original candidates with the closest indexes and then extract the targets by the extending strategy.

5 Conclusion

In this paper, we propose a Shared-Private Representation Model (SPRM) with coarse-to-fine extraction for target sentiment analysis. To encode the information of the two sub-tasks of target sentiment analysis, a Shared-Private Network has been proposed to learn shared features as well as private features. Moreover, we designed a coarse-to-fine extraction algorithm, which extracts targets without thresholds and adopts an extending strategy for better extracting target phrases. Experiments on three benchmark datasets show the effectiveness of SPRM.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (Grants no. 61772568), Guangdong Basic and Applied Basic Research Foundation (Grant no. 2019A1515012029), and Guangdong Special Support Program.

References

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2018. Zero-resource multilingual model transfer: Learning what to share.

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*.

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. *arXiv preprint arXiv:1906.06906*.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. *arXiv preprint arXiv:1906.03820*.

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

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6714–6721.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019b. Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*.

Peiqin Lin, Meng Yang, and Jianhuang Lai. 2019. Deep mask memory network with semantic dependency and context moment for aspect level sentiment classification. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5088–5094.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. Doer: Dual cross-shared rnn for aspect term-polarity co-extraction. *arXiv preprint arXiv:1906.01794*.

Dehong Ma, Sujian Li, and Houfeng Wang. 2018. Joint learning for targeted sentiment analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4737–4742.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1654, Seattle, Washington, USA. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601*.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 612–621, Lisbon, Portugal. Association for Computational Linguistics.

Yan Zhou, Longtao Huang, Tao Guo, Jizhong Han, and Songlin Hu. 2019. A span-based joint model for opinion target extraction and target sentiment classification. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5485–5491. AAAI Press.