A Multi-Task Incremental Learning Framework with Category Name Embedding for Aspect-Category Sentiment Analysis

Zehui Dai*, Cheng Peng, Huajie Chen, Yadong Ding
NLP Group, Gridsum
{daizehui,pengcheng01,chenhuajie,dingyadong}@gridsum.com

Abstract

(T)ACSA tasks, including aspect-category sentiment analysis (ACSA) and targeted aspect-category sentiment analysis (TACSA), aims at identifying sentiment polarity on predefined categories. Incremental learning on new categories is necessary for (T)ACSA real applications. Though current multi-task learning models achieve good performance in (T)ACSA tasks, they suffer from catastrophic forgetting problems in (T)ACSA incremental learning tasks. In this paper, to make multi-task learning feasible for incremental learning, we proposed Category Name Embedding network (**CNE**-net). We set both encoder and decoder shared among all categories to weaken the catastrophic forgetting problem. Besides the origin input sentence, we applied another input feature, i.e., category name, for task discrimination. Our model achieved state-of-the-art on two (T)ACSA benchmark datasets. Furthermore, we proposed a dataset for (T)ACSA incremental learning and achieved the best performance compared with other strong baselines.

1 Introduction

Sentiment analysis has become an increasingly popular natural language processing (NLP) task in academia and industry. It provides real-time feedback on consumer experience and their needs, which helps producers to offer better services. To deal with the presence of multiple categories in one document, **(T)ACSA** tasks, including aspect-category sentiment analysis (**ACSA**) and targeted aspect-category sentiment analysis (**TACSA**), were introduced.

The main purpose for ACSA task is to identify sentiment polarity (i.e. positive, neutral, negative and none) of an input sentence upon specific predefined categories (Mohammad et al., 2018; Wu et al., 2018). For example, as shown in Table 1, giving an input sentence “Food is always fresh and hot-ready to eat, but it is too expensive.” and predefined categories {*food*, *service, price*, *ambience* and *anecdotes/miscellaneous*}, the sentiment of category *food* is positive, the polarity regarding to category *price* is negative, while is none for others. In this task, the models should capture both explicit expressions and implicit expressions. For example, the phrase “too expensive” indicates the negative polarity in the *price* category, without a direct indication of “price”.

In order to deal with ACSA with both multiple categories and multiple targets, TACSA task was introduced (Saeidi et al., 2016) to analyze sentiment polarity on a set of predefined target-category pairs. An example is shown in Table 1, given targets “restaurant-1” and “restaurant-2”, in the case “I like restaurant-1 because it’s cheap, but restaurant-2 is too expansive”, the category *price* for target “restaurant-1” is positive, but is negative for target “restaurant-2”, while is none for other target-category pairs. A mathematical definition for (T)ACSA is given as follows: giving a sentence s as input, a predefined set of targets T and a predefined set of aspect categories A, a model predicts the sentiment polarity y for each target-category pair $\{(t, a) : t \in T, a \in A\}$. For ACSA task, there is only one target t in all (t, a) categories. In this paper, in order to simplify the expression in TACSA, we use predefined categories, which is short for predefined target-category pairs.

Multi-task learning, with shared encoders but individual decoders for each category, is an approach to analyze all the categories in one sample simultaneously for (T)ACSA (Akhtar et al., 2018; Schmitt et al., 2018). Compared with single-task ways (Liang et al., 2019), multi-task approaches utilize category-specific knowledge in training signals from each task and get better performance. However, current multi-task models still suffer from a

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6955–6965,
November 16–20, 2020. ©2020 Association for Computational Linguistics

Task	Sentence	Labels
ACSA	Food is always fresh and hot-ready to eat, but it is too expensive	(food,positive), (service, none), (price, negative), (ambience, none) (anecdotes/miscellaneous, none)
TACSA	I like restaurant-1 because it's cheap, but restaurant-2 is too expansive.	(restaurant-1-general, none), (restaurant-1-price,positive), (restaurant-1-location, none), (restaurant-1-safety,none), (restaurant-2-general, none), (restaurant-2-price,negative), (restaurant-2-location, none), (restaurant-2-safety,none)

Table 1: Example and gold standard for (T)ACSA examples.

lack of features such as category name (Meisheri and Khadilkar, 2018). Models with category name features encoded in the model may further improve the performance.

On the other hand, the predefined categories in (T)ACSA task make the application in new categories inflexible, as for (T)ACSA applications, the number of categories maybe varied over time. For example, *fuel consumption, price level, engine power, space* and so on are **source categories** to be analyzed in the gasoline automotive domain. For electromotive domain, source categories in the automotive domain will still be used, while new **target category** such as *battery duration* should also be analyzed. Incremental learning is a way to solve this problem. Therefore, it is necessary to propose an incremental learning task and an incremental learning model concerned with new category for (T)ACSA tasks.

Unfortunately, in the current multi-task learning (T)ACSA models, the encoder is shared but the decoders for each category are individual. This parameter sharing mechanism results in only the shared encoder and target-category-related decoders are finetuned during the finetuning process, while the decoder of source categories remains unchanged. The finetuned encoder and original decoder of source categories may cause catastrophic forgetting problem in the origin categories. For real applications, high accuracy is excepted in source categories and target categories. Based on the previous researches that decoders between different tasks are usually modeled by mean regularization (Evgeniou and Pontil, 2004) , an idea comes up to further make the decoders the same by sharing the decoders in all categories to decrease the catastrophic forgetting problem. But here raises another question, how to identify each category in the encoder and decoder shared network? In our approach, we solve the category discrimination problem by the input category name feature.

In this paper, we proposed a multi-task category name embedding network (**CNE**-net). The multi-task learning framework makes full use of training signals from all categories. To make it feasible for incremental learning, both encoder and decoders for each category are shared. The category names were applied as another input feature for task discrimination. We also present a new task for (T)ACSA incremental learning. In particular, our contribution is three-folded:

(1) We proposed a multi-task **CNE**-net framework with both encoder and decoder shared to weaken catastrophic forgetting problem in multi-task learning (T)ACSA model.

(2) We achieved state-of-the-art on the two (T)ACSA datasets, SemEval14-Task4 and Sentihood.

(3) We proposed a new task for incremental learning in (T)ACSA. By sharing both encoder layers and decoder layers of all the tasks, we achieved better results compared with other baselines both in source categories and in the target category.

2 Related Work

2.1 Aspect-category Sentiment Analysis

(T)ACSA task is to predict sentiment polarity on a set of predefined categories. It is able to analyze sentiment in an end-to-end way with explicit expressions or implicit expressions (Mohammad et al., 2018; Wu et al., 2018). The earliest works most concerned on feature engineering (Zirn et al., 2011; Wiebe, 2012; Wagner et al., 2014). Subsequently, Nguyen and Shirai (2015); Wang et al. (2017); Meisheri and Khadilkar (2018) applied neural network models to achieve higher accuracy. Ma et al. (2018) then involved commonsense knowledge as additional features. The current approaches consist of multi-task models (Akhtar et al., 2018; Schmitt et al., 2018), which analyze all the categories simultaneously in one sample to make full use of all the features and labels in the training sample, and single-task models that treat one category in one sample (Jiang et al., 2019).

2.2 Multi-Task Learning

Multi-task learning(MTL) utilizes all the related tasks by sharing the commonalities while learning individual features for each sub-task. MTL has been proven to be effective in many NLP tasks, such as information retrieval (Liu et al., 2015), machine translation (Dong et al., 2015), and semantic role labeling (Collobert and Weston, 2008). For ACSA task, Schmitt et al. (2018) applied MTL framework with a shared LSTM encoder and individual decoder classifiers for each category. The multiple aspects in MTL were handled by constrained attention networks with orthogonal and sparse regularization (Hu et al., 2019).

2.3 Incremental Learning

Incremental learning was inspired by adding new abilities to a model without having to retrain the entire model. For example, Doan and Kalita (2016) presented several random forest models to perform sentiment analysis on customers' reviews. Many domain adaptation approaches utilizing transfer learning suffer from "catastrophic forgetting" problem (French and Chater, 2002). To solve this problem, Rosenfeld and Tsotsos (2017) proposed an incremental learning Deep-Adaption-Network that constrains newly learned filters to be linear combinations of existing ones.

To the best of our knowledge, for (T)ACSA task, few researches concerned with incremental learning in new categories. In this paper, we proposed a (T)ACSA incremental learning task and the **CNE**-net model to solve this problem in a multi-task learning approach with a shared encoder and shared decoders. We also apply category name for task discrimination.

3 Datasets

This section describes the benchmark datasets we used to evaluate our model, the incremental learning task definition, the methodology to prepare the incremental learning dataset, and the evaluation metric.

3.1 Evaluation Benchmark Datasets

We evaluated the performance of the **CNE**-net model on two benchmark datasets, i.e., ACSA task on SemEval-2014 Task4 (Pontiki et al., 2014) and TACSA task on SentiHood (Saeidi et al., 2016).

The **ACSA task** was evaluated on SemEval-2014 Task4, a dataset on restaurant reviews. Our model provides a joint solution for sub-task 3 (Aspect Category Detection) and sub-task 4 (Aspect Category Sentiment Analysis). The sentiment polarities are $y \in Y = \{$positive, neutral, negative, conflict and none$\}$, and the categories are $a \in A = \{$*food*, *service*, *price*, *ambience* and *anecdotes/miscellaneous*$\}$. The conflict label indicates both positive and negative sentiment is expressed in one category (Pontiki et al., 2014).

The **TACSA task** was evaluated on the Sentihood dataset, which describes locations or neighborhoods of London and was collected from question answering platform of Yahoo. The sentiment polarities are $y \in Y = \{$positive, negative and none$\}$, the targets are $t \in T = \{$Location1, and Location2$\}$, and the aspect categories are $a \in A = \{$*general*, *price*, *transit-location*, and *safety*$\}$.

3.2 Evaluation Transfer Learning Datasets

Besides evaluating the model on existing (T)ACSA tasks, we also proposed incremental learning tasks for (T)ACSA[1] in new category based on SemEval-2014 Task4 and Sentihood dataset, respectively.

Firstly, we split the categories into source categories and target categories. For ACSA task, the source categories are $\{$*food*, *price*, *ambience* and *anecdotes/miscellaneous*$\}$, while the target category is $\{$*service*$\}$. For TACSA task, the source categories are $\{$*general*, *transit-location*, and *safety*$\}$, while the target category is $\{$*price*$\}$. This was considered by the amount of data with positive/negative/neutral polarity in this category, as well as the sense of this category for real applications.

[1] The dataset can be found at `https://github.com/flak300S/emnlp2020_CNE-net`.

origin ACSA sample	{“text”: “The only thing more wonderful than the food is the service.”, ”sentiment”: {“food”: “Positive”, “service”: “Positive”, “price”: None, “ambience”: None, “anecdotes/miscellaneous”: None } }
ACSA Sample-Source	{“text”: “The only thing more wonderful than the food is the service.”, ”sentiment”: {“food”: “Positive”, “price”: None, “ambience”: None, “anecdotes/miscellaneous”: None } }
ACSA Sample-Target	{“text”: “The only thing more wonderful than the food is the service.”, ”sentiment”: {“service”: “Positive” } }

Table 2: An example for generating ACSA incremental learning task.

Secondly, we prepare training, validation and testing data for incremental learning task by independently splitting the origin training data, validation data and test data into source-category data (**Sample-Source**) containing label only in source categories and target-category data (**Sample-Target**) with target-category label only. For example, as shown in Table 2, in ACSA task, the origin labels {*food*: positive, *service*:positive, *price*:none, *ambience*:none, *anecdotes/miscellaneous*:none} were transformed to {*food*: positive, *price*:none, *ambience*:none, *anecdotes/miscellaneous*:none} in Sample-Source and {*service*:positive} in Sample-Target. The input sentences were kept the same as origin dataset. For other researches to investigate the influence of target-category training data amount quantitatively, we also created incremental learning data by combining all the Sample-Source and sampled Sample-Target. The sampling rate is a range from 0.0 to 1.0.

In this paper, the ACSA incremental learning dataset is created from SemEval14-Task ACSA dataset, and it is called SemEval14-Task-*inc*. The TACSA incremental learning dataset is created from Sentihood TACSA dataset, and it is called Sentihood-*inc*.

3.3 Evaluation Metrics

We evaluated the aspect category extraction (to determine whether the sentiment is none for each category) and sentiment analysis (to predict the sentiment polarity) on the two datasets. For aspect category extraction evaluation, we applied the probability $1 - p$ as the not none probability for each category, where p is the probability of the “none” class in this category. The evaluation metric is the same as Sun et al. (2019). For the origin SemEval-14 Task4 dataset, we use Micro-F1 for category extraction evaluation and accuracy for sentiment analysis evaluation. For the origin Sentihood dataset, we use Macro-F1, strict accuracy, and area-under-curve(AUC) for category extraction evaluation while use AUC, and strict accuracy for sentiment analysis evaluation. When evaluating the incremental learning task, we use the F1 metric (Micro-F1 for SemEval-14 and Macro-F1 for Sentihood) for category extraction and accuracy for sentiment analysis.

4 Approach

In this section, we describe the architecture of **CNE**-net for (T)ACSA task. In BERT classification tasks, the typical approach is feeding sentence “[CLS]tokens in sentence[SEP]” into the model, while the token “[CLS]” is used as a feature for classification. In order to encode category names into BERT model, as well as analyze sentiment polarity of all the categories simultaneously, we made two significant differences from the original BERT, one on the encoder module and another on the decoder module.

4.1 Encoder with Category Name Embedding

In order to get a better category name embedding, as well as to make it feasible for incremental learning cross categories, the category names are encoded into the model, along with the origin sentence like “[CLS] sentence words input [SEP] category1 input [SEP] category2 input [SEP]...[SEP] categoryN input[SEP]”, as shown in the BERT encoder module in Figure 1. In ACSA task, the category names are “{food, service, price, ambiance, and anecdotes/miscellaneous}”, while in TACSA task, the category names are “{location-1 general, location-1 price, location-1 transit-location, location-1 safety, location-2 general, location-2 price, location-2 transit-location, and location-2 safety}”.

We mark output states of the BERT encoder as follows: the hidden state of [CLS] $\vec{h}_{[CLS]} \in \mathbb{R}^d$, the hidden states of words in origin sentences

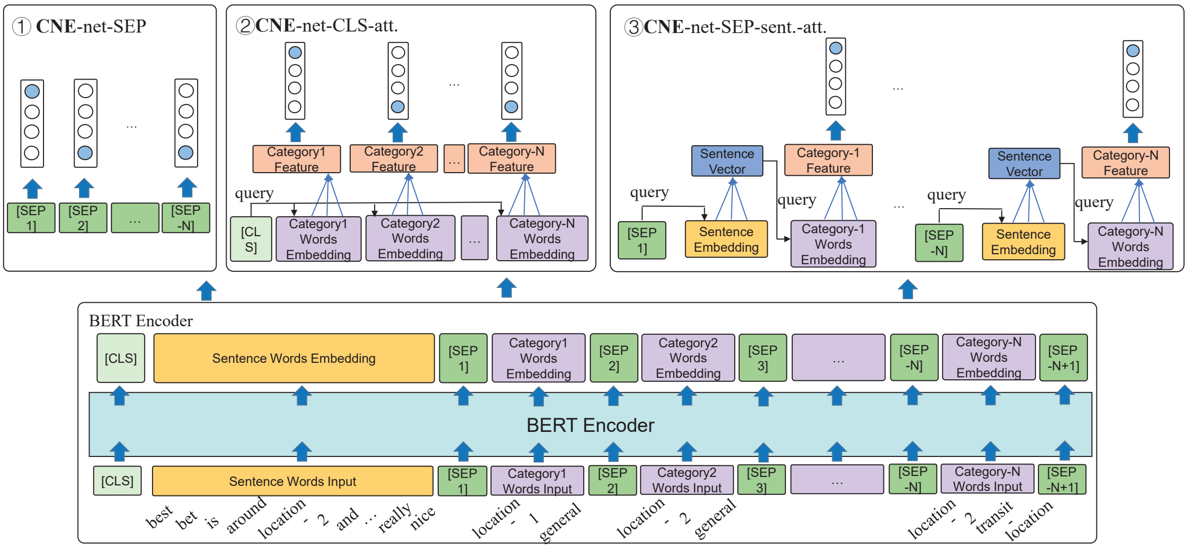

Figure 1: **CNE**-net model architecture

$\mathbf{H}_{sent} \in \mathbb{R}^{L_{sent} \times d}$, the hidden states of separators $\mathbf{H}_{[SEP]} \in \mathbb{R}^{n_{cat} \times d}$, and the hidden states of category words $\mathbf{H}_{cat-i} \in \mathbb{R}^{L_{cat-i} \times d}$ for the i-th category ($0 < i \leq n_{cat}$), where L_{sent} is the length of the input sentence, d is the dimension of hidden states, n_{cat} is the number of categories feed into the model, and L_{cat-i} is the length of the i-th category input words.

4.2 Multi-Task Decoders

We proposed three types of decoder for (T)ACSA task, as shown in Figure 1①,② and ③. These decoders are multi-label classifiers, which apply a softmax classifier for sentiment analysis in each category.

Type 1, **CNE**-net-SEP, as shown in Figure 1①, the separator token $\vec{h}_{[SEP-i]}$ is applied as feature representation for sentiment polarity analysis in each category directly. The probability for each polarity in category *i* is calculated as follows where $\vec{h} = \vec{h}_{[SEP-i]}$:

$$\vec{f_i} = \mathbf{W}_i \cdot \vec{h} + \vec{b_i}; \vec{p_i} = softmax(\vec{f_i}) \quad (1)$$

where $\vec{f_i} \in \mathbb{R}^s$ is the output logits for category i, $\vec{p_i} \in \mathbb{R}^s$ is the output probability for category i, $\mathbf{W}_i \in \mathbb{R}^{d \times s}$ and $\vec{b_i} \in \mathbb{R}^s$ are randomly initialized parameters to be trained, and s is the number of sentiment classes. $s = 5$ for $\{$positive, neutral, negative, conflict and none$\}$ in SemEval14-Task4, while $s = 3$ for $\{$positive, negative and none$\}$ in Sentihood dataset. In our approach, $\mathbf{W}_1 = \mathbf{W}_2 = ... = \mathbf{W}_{n_{cat}}$ and $\vec{b}_1 = \vec{b}_2 = ... = \vec{b}_{n_{cat}}$.

Type 2, **CNE**-net-CLS-att., in order to get content-aware category embedding vector, we applied attention mechanism with $\vec{h}_{[CLS]}$ serves as query vector, and $\mathbf{H}_{cat-i}$ serves as both key and value matrix, as shown in Figure 1②. The category embedding vector $\vec{e}_{cat-i}$ for the *i*-th category is as follows:

$$\vec{e}_{cat_i} = softmax(\vec{h}_{[CLS]} \cdot \mathbf{H}_{cat-i}) \cdot \mathbf{H}_{cat-i} \quad (2)$$

The probability for category *i* in type 2 is calculated following equation(1) where $\vec{h} = \vec{e}_{cat_i}$.

Type 3, **CNE**-net-SEP-sent.-att. applied attention mechanism for both sentence embedding and category name embedding. As it is shown in Figure 1③. Firstly, sentence vector correlated with the *i*-th category is calculated by attention with separator embedding $\vec{h}_{[SEP-i]}$ serving as query, and sentence embedding $\mathbf{H}_{sent}$ serving as key and value matrix. Sentence vector $\vec{h}_{sent-i}$ correlated with the *i*-th category is as follows:

$$\vec{h}_{sent-i} = softmax(\vec{h}_{[SEP-i]} \cdot \mathbf{H}_{sent}) \cdot \mathbf{H}_{sent} \quad (3)$$

Secondly, similar to that in type 2, the category embedding vector $\vec{e}_{cat-i}$ for the *i*-th category calculated by attention mechanism is as follows:

$$\vec{e}_{cat_i} = softmax(\vec{h}_{sent-i} \cdot \mathbf{H}_{cat-i}) \cdot \mathbf{H}_{cat-i} \quad (4)$$

The probability for for category *i* in type 3 is calculated following equation(1) where $\vec{h} = \vec{e}_{cat_i}$.

4.3 Model Training

The **CNE**-net multi-task framework was trained in an end-to-end way by minimizing the sum of cross-entropy loss of all the categories. We employed L_2 regularization to ease over-fitting. The loss function is given as follows:

$$L = -\frac{1}{|D|}\sum_{x,y\in D}\sum_{i=1}^{N}\vec{y}_i \cdot \log \vec{p}_i(x;\theta) + \frac{\lambda}{2}||\theta||_2 \tag{5}$$

where D is the training dataset, N is the number of categories, Y is the sentiment classes $Y = \{positive, neutral, negative, conflict, none\}$ (*neutral* and *conflict* is not included in TACSA task), $\vec{y}_i \in \mathbb{R}^{|Y|}$ is the one-hot label vector for the i-th category with true label marked as 1 and others marked as 0, $\vec{p}_i(x;\theta)$ is the probability for the i-th category, and λ is the L_2 regularization weight. Besides L_2 regularization, we also employed dropout and early stopping to ease over-fitting.

During training incremental learning models, we follow the workflow of the incremental learning application. We firstly train a source-category model with the Sample-Source training data. Then finetuned the source-category model with Sample-Target training data to get incremental learning model.

5 Experiments

5.1 Experiment Settings

The pretrained uncased BERT-base[2] was used as the encoder in **CNE**-net. The number of Transformer blocks is 12, the number of self-attention heads is 12, and the hidden layer size in each self-attention head is 64. The total amount of parameters in BERT encoder is about 110M. The dropout ratio is 0.1 during training, the traning epochs is 10, and the learning rate is 5e-5 with a warm-up ratio of 0.25.

5.2 Compared Methods

We compare the performance of our model with some state-of-the-art models.

For ACSA task:

- XRCE (Brun et al., 2014): a hybrid classifier based on linguistic features.
- NRC-Canada (Kiritchenko et al., 2014): several binary one-vs-all SVM classifiers for this multi-class multi-label classification problem.
- AT-LSTM and ATAE-LSTM (Wang et al., 2016): a LSTM attention framework with aspect word embeddings concatenated with sentence word embeddings.
- BERT-pair-QA-B (Sun et al., 2019): a question answering and natural language inference model based on BERT.
- Multi-task framework (MTL) (Schmitt et al., 2018): a LSTM multi-task learning framework with an individual attention head for each category. To better compare our model with this approach, we changed the encoder to BERT-base.

For TACSA task:

- LR (Saeidi et al., 2016): a logistic regression classfier with linguistic features.
- LSTM-final (Saeidi et al., 2016): a BiLSTM encoder with final states served as feature representation.
- LSTM+TA+SA (Ma et al., 2018): a BiLSTM encoder with complex target-level and sentence-level attention mechanisms.
- SenitcLSTM (Ma et al., 2018): LSTM+TA+SA model upgraded by introducing external knowledge.
- Dmu-Entnet (Liu et al., 2018): model with delayed memory update mechanism to track different targets.
- Recurrent Entity Network (REN) (Ye and Li, 2020): a recurrent entity memory network that employs both word-level information and sentence-level hidden memory for entity state tracking.

In TACSA task, besides these models, we also compared our model with the BERT-pair-QA-B model and MTL model mentioned in ACSA comparison methods.

5.3 Main Results

The performances of compared methods and three types of **CNE**-net are shown in Table 3 (ACSA task) and Table 4 (TACSA task). All the models with BERT encoder (QA-B, MTL and our **CNE**-net) achieved better performance compared with models without BERT encoder (XRCE, NCR-Canada, AT-LSTM, ATAE-LSTM, SenitcLSTM, Dmu_entnet, and REN). Our **CNE**-net performs better compared with QA-B and MTL framework

[2] https://storage.googleapis.com/bert_models /2018_10_18/uncased_L-12_H-768_A-12.zip

Model	Category Extraction			Sentiment Analysis		
	P	R	F	binary	3-way	4-way
XRCE (Brun et al., 2014)	83.23	81.37	82.29	-	-	78.1
NRC-Canada (Kiritchenko et al., 2014)	91.04	86.24	88.58	-	-	82.9
AT-LSTM (Wang et al., 2016)	-	-	-	89.6	83.1	-
ATAE-LSTM (Wang et al., 2016)	-	-	-	89.9	84.0	-
QA-B (Sun et al., 2019)	93.04	89.95	91.47	95.6	89.9	85.9
MTL	91.87	90.44	91.15	95.0	88.8	85.3
CNE-net-SEP (ours)	92.26	90.73	91.49	95.8	90.2	86.3
CNE-net-CLS-att. (ours)	93.37	90.93	91.98	96.1	91.0	87.0
CNE-net-SEP-sent.-att. (ours)	93.76	90.83	**92.27**	**96.4**	**91.3**	**87.1**

Table 3: Performance on SemEval-14 Task4, ACSA task. ("-" means not reported.)

Model	Category Extraction			Sentiment Analysis	
	Acc.	F_1	AUC	*Acc.*	AUC
LR (Saeidi et al., 2016)	-	39.3	92.4	87.5	90.5
LSTM-final (Saeidi et al., 2016)	-	68.9	89.8	82.0	85.4
LSTM+TA+SA (Ma et al., 2018)	66.4	76.7	-	86.8	-
SenticLSTM (Ma et al., 2018)	67.4	78.2	-	89.3	-
Dmu-Entnet (Liu et al., 2018)	73.5	78.5	94.4	91.0	94.8
REN (Ye and Li, 2020)	75.7	80.4	96.0	92.5	95.9
QA-B (Sun et al., 2019)	79.2	87.9	97.1	93.3	97.0
MTL	80.4	88.4	97.6	93.6	97.1
CNE-net-SEP (ours)	80.2	88.1	97.6	93.4	97.3
CNE-net-CLS-att. (ours)	80.4	88.8	97.8	93.8	97.4
CNE-net-SEP-sent.-att. (ours)	**80.8**	**89.4**	**97.9**	**94.0**	**97.5**

Table 4: Performance on Sentihood, TACSA task. ("-" means not reported.)

in both ACSA and TACSA tasks. QA-B is a single-task approach, which each category is trained independently. Our **CNE**-net is a multi-task learning framework. It performs better than QA-B by using shared semantic features and sentiment labels in all the categories. **CNE**-net also performs better compared with the MTL model since it encodes the category names as additional features to generate the representation of each category.

Our **CNE**-net-SEP-sent.-att. model achieves state-of-the-art on all the evaluation metrics in both SemEval14-Task4 and Sentihood dataset. The improved extraction F_1 is 0.0080 in the SemEval14-Task4 (increased from 0.9147 in QA-B to 0.9227 in **CNE**-net-SEP-sent.att.), while it is 0.010 in the Sentihood dataset (increased from 0.884 in MTL to 0.894 in **CNE**-net-SEP-sent.att.). The accuracy metrics for sentiment analysis in the SemEval14-Task4 are binary, 3-way and 4way, which refers to accuracy with positive/negative (binary), positive/neutral/negative (3-way) and positive/neutral/negative/conflict (4-way). The improvement of sentiment classification accuracy is 0.012 in SemEval14-Task4 (4-way setting, increased from 0.859 in QA-B to 0.871 in **CNE**-net-SEP-sent.att.), while is 0.004 in the Sentihood dataset (increased from 0.971 in MTL to 0.975 in **CNE**-net-SEP-sent.att.).

CNE-net-SEP uses [SEP] as a feature representation for sentiment classification. It performs the poorest among all three types of **CNE**-net since representation from only [SEP] token does not make full use of sentence information and category information. **CNE**-net-CLS-att. uses [CLS] as sentence representation and applies attention mechanism to build the relationship between sentence representation and the category name hidden states to get sentiment classification feature and achieve better performance. The **CNE**-net-SEP-sent.-att. uses attention twice. The first one is to build category-name-aware sentence embeddings for each category with [SEP] as query and sentence hidden states matrix as key and value, while the second one is to apply each category-name-aware sentence embedding to generate category representation like what we do in **CNE**-net-CLS-att.. This category-name-aware sentence embedding and the sentence-aware category embedding makes it per-

Model	SemEval14-Task4-*inc*				Sentihood-*inc*			
	$extra.$		$senti.$		$extra.$		$senti.$	
	$mix.$	$incre.$	$mix.$	$incre.$	$mix.$	$incre.$	$mix.$	$incre.$
AE-LSTM	85.3	85.0	85.2	85.9	86.3	86.5	84.4	84.5
ATAE-LSTM	85.6	85.2	85.4	86.0	86.6	86.9	84.6	84.7
Dmu-Entnet	-	-	-	-	87.9	88.0	85.4	85.8
QA-B	92.2	92.5	91.9	92.0	93.7	93.6	90.6	91.0
MTL	92.5	92.6	92.4	92.5	93.8	93.7	90.8	91.4
CNE-SEP(ours)	92.9	92.7	92.5	92.8	94.5	94.8	91.2	91.6
CNE-net-CLS-sent.(ours)	93.0	92.8	92.7	93.0	94.8	95.0	91.6	91.7
CNE-net-SEP-sent.-att. (ours)	**93.6**	**93.7**	**93.0**	**93.2**	**95.2**	**95.4**	**91.9**	**92.0**

Table 5: Extraction F_1 and sentiment accuracy in target category of incremental learning.

Model	SemEval14-Task4-*inc*				Sentihood-*inc*			
	$extra.$		$senti.$		$extra.$		$senti.$	
	$mix.$	$incre.$	$mix.$	$incre.$	$mix.$	$incre.$	$mix.$	$incre.$
AE-LSTM	83.6	83.4	78.3	77.9	82.3	81.5	85.1	84.0
ATAE-LSTM	83.7	83.5	78.7	78.0	82.6	81.6	85.6	85.0
Dmu-Entnet	-	-	-	-	83.2	82.3	85.8	85.2
QA-B	90.0	89.2	84.4	83.5	85.2	84.2	91.7	90.7
MTL	89.8	69.8↓	84.5	82.3	87.0	75.7↓	92.2	91.0
CNE-SEP(ours)	90.9	90.1	84.8	84.5	87.2	85.8	92.6	91.6
CNE-net-CLS-sent.(ours)	91.2	91.1	85.4	85.0	87.5	86.1	93.0	91.9
CNE-net-SEP-sent.-att. (ours)	**91.6**	**91.3**	**85.5**	**85.4**	**87.7**	**86.3**	**93.2**	**92.3**

Table 6: Extraction F_1 and sentiment accuracy in source categories of incremental learning.

form the best in the three types of **CNE**-net.

5.4 Incremental Learning Results

This section describes the performance in the incremental learning task. We trained the model following incremental learning workflow, as mentioned in section 4.3. We compared the results between mix-training (short as $mix.$) (mixing Sample-Source and Sample-Target) and incremental learning (short as $incre.$), for both extraction F_1 and sentiment accuracy.

Firstly, we compare the performance in target category, i.e. aspect category extraction F_1 (short as $extra.$) and sentiment analysis accuracy (short as $senti.$) from mix-training process and incremental learning. As the target category performance shown in Table 5, there is no significant difference between mix-training and incremental learning for both aspect extraction and sentiment analysis. For example, in SemEval14-Task-*inc*, the extraction F_1 and sentiment accuracy of **CNE**-net-SEP-sent.-att. are 0.936 and 0.930 respectively in mix-training, while they are 0.937 and 0.932 respectively in incremental learning. In Sentihood-*inc*, the extraction F_1 and sentiment accuracy of **CNE**-net-SEP-sent.-att. are 0.952 and 0.919 respectively in mix-training, while they are 0.954 and 0.920 respectively in incremental learning. This indicates incremental learning does not decrease the performance in the target category. Our **CNE**-net-SEP-sent.-att. performs the best in all the models.

Secondly, we compare aspect extraction and sentiment analysis performance in source categories after incremental learning, since both source categories and target categories requires high accuracy. The extraction F_1 and sentiment accuracy of source categories after the incremental learning process as well as in the mix-training process are shown in Table 6. There is no significant difference in sentiment accuracy of source categories after training with incremental learning data. For example, in SemEval14-Task-*inc*, sentiment accuracy of **CNE**-net-SEP-sent.-att. is 0.855 in mix-training, while it is 0.854 in incremental learning. This is probably because of the similar sentiment features between categories, in which the fine-tuning process does not make a great difference.

However, for category extraction, MTL suffers from catastrophic forgetting after fine-tuning. In SemEval14-Task4-*inc*, extraction F_1 of MTL model of source categories decreases from 0.898 in mix-training to 0.698 after incremental learning,

CNE-net-SEP-sent.-att.	SemEval14-Task4-*inc*				Sentihood-*inc*			
	Source Categories		Target Category		Source Categories		Target Category	
	extra.	*senti.*	*extra.*	*senti.*	*extra.*	*senti.*	*extra.*	*senti.*
shared decoder	91.3	85.4	93.7	93.2	86.3	92.3	95.4	92.0
unshared decoder	84.2↓	84.0	93.4	93.0	79.6↓	91.5	94.9	91.6

Table 7: Extraction F_1 and sentiment accuracy after incremental learning of **CNE**-net-SEP-sent.-att. with shared and unshared decoder.

while in Sentihood-*inc*, F_1 metric of MTL model of source categories decreases from 0.870 in mix-training to 0.757 after incremental learning. Fortunately, the QA-B model, as well as our **CNE**-nets, suffer less from this problem. In SemEval14-Task4-*inc*, extraction F_1 metric of **CNE**-SEP-sent.-att. is 0.913 in source categories after fine-tuning, while it is 0.916 in mix-training. In Sentihood-*inc*, extraction F_1 of **CNE**-SEP-sent.-att. is 0.863 in source categories after fine-tuning, while it is 0.877 in mix-training.

5.5 Discussion

We have confirmed the effectiveness of **CNE**-nets for (T)ACSA tasks and (T)ACSA incremental learning tasks. However, there remains a question, why our model suffers less from catastrophic forgetting in incremental learning?

To answer this question, we compare the incremental learning performance of our **CNE**-net-SEP-sent.-att. with a similar model but the decoders in each category are unshared with $\mathbf{W}_1 \neq \mathbf{W}_2 \neq ... \neq \mathbf{W}_{n_{cat}}$ and $\vec{b}_1 \neq \vec{b}_2 \neq ... \neq \vec{b}_{n_{cat}}$ (**CNE**-net-SEP-sent.-att.-unshared) in equation (1) and the results are shown in Table 7. There is no significant difference in target category between the model with shared decoders and the model with unshared decoders, indicating both shared and unshared model is able to get enough feature for category extraction and sentiment analysis in target category. However, it is more important that, in **CNE**-net-SEP-sent.-att.-unshared, the extraction F_1 suffers from a sudden decrease. In SemEval14-Task4-*inc*, extraction F_1 decreases from 0.913 with shared decoder to 0.842 with unshared decoder, while in Sentihood-*inc*, extraction F_1 decreases from 0.863 with shared decoder to 0.796 with unshared decoder.

We believe the decreased extraction F_1 in source categories is due to the unshared decoders for each task, which results in only shared encoder and target-category decoders are fine-tuned during the fine-tuning process. In contrast, the decoder of source categories remains unchanged. The fine-tuned encoder and original source-category decoder is the reason for the catastrophic forgetting problem in the category extraction evaluation. In our shared decoder approach, both encoders and decoders are shared and fine-tuned to weaken the catastrophic forgetting problem.

6 Conclusion

In this paper, in order to make multi-task learning feasible for incremental learning, we proposed **CNE**-net with different attention mechanisms. The category name features and the multi-task learning structure help the model achieve state-of-the-art on ACSA and TACSA tasks. Furthermore, the shared encoder and decoder layers weaken catastrophic forgetting in the incremental learning task. We proposed a task for (T)ACSA incremental learning and achieved the best performance with **CNE**-net compared with other strong baselines. Further research may be concerned with zero-shot learning on new categories.

References

Md. Shad Akhtar, Deepanway Ghosal, Asif Ekbal, and Pushpak Bhattacharyya. 2018. A multi-task ensemble framework for emotion, sentiment and intensity prediction. *CoRR*, abs/1808.01216.

Caroline Brun, Diana Nicoleta Popa, and Claude Roux. 2014. XRCE: hybrid classification for aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 838–842.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 160–167.

Tri Doan and Jugal K. Kalita. 2016. Sentiment analysis of restaurant reviews on yelp with incremental learning. In *15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016,* *Anaheim, CA, USA, December 18-20, 2016*, pages 697–700.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1723–1732.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 109–117. ACM.

Robert M. French and Nick Chater. 2002. Using noise to compute error surfaces in connectionist networks: A novel means of reducing catastrophic forgetting. *Neural Computation*, 14(7):1755–1769.

Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2019. CAN: constrained attention networks for multi-aspect sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4600–4609.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6279–6284.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 437–442.

Bin Liang, Jiachen Du, Ruifeng Xu, Binyang Li, and Hejiao Huang. 2019. Context-aware embedding for targeted aspect-based sentiment analysis. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4678–4683.

Fei Liu, Trevor Cohn, and Timothy Baldwin. 2018. Recurrent entity networks with delayed memory update for targeted aspect-based sentiment analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 278–283.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 912–921.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5876–5883.

Hardik Meisheri and Harshad Khadilkar. 2018. Learning representations for sentiment classification using multi-task framework. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2018, Brussels, Belgium, October 31, 2018*, pages 299–308.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 1–17.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2509–2514.

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@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35.

Amir Rosenfeld and John K. Tsotsos. 2017. Incremental learning through deep adaptation. *CoRR*, abs/1705.04228.

Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. 2016. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *COLING 2016, 26th International Conference on Computational Linguistics,* *Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1546–1556.

Martin Schmitt, Simon Steinheber, Konrad Schreiber, and Benjamin Roth. 2018. Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1109–1114.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 380–385.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. DCU: aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 223–229.

Bo Wang, Maria Liakata, Arkaitz Zubiaga, and Rob Procter. 2017. Tdparse: Multi-target-specific sentiment recognition on twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 483–493.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 606–615.

Janyce Wiebe. 2012. Subjectivity word sense disambiguation. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, WASSA@ACL 2012, July 12, 2012, Jeju Island, Republic of Korea*, page 2.

Chuhan Wu, Fangzhao Wu, Junxin Liu, Zhigang Yuan, Sixing Wu, and Yongfeng Huang. 2018. Thu_ngn at semeval-2018 task 1: Fine-grained tweet sentiment intensity analysis with attention CNN-LSTM. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018, New Orleans, Louisiana, USA, June 5-6, 2018*, pages 186–192.

Zhihao Ye and Zhiyong Li. 2020. A variant of recurrent entity networks for targeted aspect-based sentiment analysis. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2268–2274. IOS Press.

Cäcilia Zirn, Mathias Niepert, Heiner Stuckenschmidt, and Michael Strube. 2011. Fine-grained sentiment analysis with structural features. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 336–344.