Multi-task Coupled Attentions for Category-specific Aspect and Opinion Terms Co-extraction

Wenya Wang†‡ Sinno Jialin Pan† Daniel Dahlmeier‡
†Nanyang Technological University, Singapore
‡SAP Innovation Center Singapore
†{wa0001ya, sinnopan}@ntu.edu.sg, ‡{d.dahlmeier}@sap.com

Abstract

In aspect-based sentiment analysis, most existing methods either focus on aspect/opinion terms extraction or aspect terms categorization. However, each task by itself only provides partial information to end users. To generate more detailed and structured opinion analysis, we propose a finer-grained problem, which we call category-specific aspect and opinion terms extraction. This problem involves the identification of aspect and opinion terms within each sentence, as well as the categorization of the identified terms. To this end, we propose an end-to-end multi-task attention model, where each task corresponds to aspect/opinion terms extraction for a specific category. Our model benefits from exploring the commonalities and relationships among different tasks to address the data sparsity issue. We demonstrate its state-of-the-art performance on three benchmark datasets.

1 Introduction

Aspect-based sentiment analysis deals with token-level predictions, aiming to provide fine-grained information. Under this branch, most work has been proposed for aspect/opinion terms extraction (Hu and Liu, 2004; Qiu et al., 2011; Wang et al., 2016), where an aspect term refers to a word or a phrase describing some feature of an entity, and an opinion term refers to the expression carrying subjective emotions. For example, in the sentence “The soup is served with nice portion, the service is prompt”, soup, portion and service are aspect terms, while nice and prompt are opinion terms. However, the above task simply extracts target terms without classifying them into different categories, which should be more useful for generating structured aspect-based opinion summaries.

On the other hand, some work focused on categorization of aspect terms. Given an extracted aspect term, early work (Carenini et al., 2005; Yu et al., 2011) applied lexicon and taxonomy-based methods to classify it to a category according to some distance measures. Semi-supervised methods have also been proposed for this problem (Zhai et al., 2010, 2011). However, this task requires the aspect terms to be extracted beforehand. Although topic models (Guo et al., 2009; Titov and McDonald, 2008a) can achieve both grouping and extraction at the same time, they mainly focused on grouping and can only identify general and coarse-grained aspect terms.

To address the limitations of existing methods and provide a more useful end-to-end analysis, we introduce a finer-grained task called category-specific aspect and opinion terms extraction, where the aspect/opinion terms need to be extracted and classified to a category from a pre-defined set. Consider the previous example, our objective is to extract and classify soup and portion as aspect terms under the “DRINKS#STYLE_OPTION” category, and service as an aspect term under the “SERVICE#GENERAL” category, similar for the opinion terms nice and prompt.

The proposed task is much more challenging because when considering different categories, the data could become extremely sparse, i.e. certain categories might only contain very few reviews. Moreover, it requires to achieve both extraction and categorization at the same time which significantly increases the difficulty compared to both tasks. Under our problem setting, existing methods are not directly applicable. Even if we modify the extraction methods to incorporate category information, the result may still be unpromising, be-
cause they fail to utilize inter-category correlations which can help address data sparsity issue. At this point, we propose a multi-task deep learning method to exploit commonalities and correlations among different tasks, where each task is defined as aspects/opinions extraction for a specific category. Inspired by (Wang et al., 2017), we model each task with coupled multi-layer attentions to extract the relations between aspect terms and opinion terms within each category. The multi-layer coupled attentions for each task/category are jointly learned in a multi-task learning manner. In summary, the contributions are two-fold: 1) We offer an end-to-end deep multi-task learning model to accomplish a finer-grained sentiment analysis task; 2) We demonstrate its state-of-the-art performance on three benchmark datasets.

2 Related Work

2.1 Fine-grained Sentiment Analysis

For the task of aspect/opinion terms extraction, there are mainly four approaches. The first approach aims to exploit syntactic dependency relations among aspect terms and opinion terms for information extraction (Hu and Liu, 2004; Popescu and Etzioni, 2005; Zhuang et al., 2006; Wu et al., 2009; Qiu et al., 2011). A second approach models the extraction of target terms as a supervised sequence labeling problem with exhaustive human-engineered features (Jin and Ho, 2009; Li et al., 2010; Jakob and Gurevych, 2010). To reduce the effort of feature engineering, a third approach aims to utilize deep learning to learn high-level features automatically (Liu et al., 2015a; Yin et al., 2016; Wang et al., 2016, 2017). However, all the above approaches only focus on opinion/aspect terms extraction without categorizing them into specific categories. A forth approach adopts topic models (Titov and McDonald, 2008b; Lu et al., 2009; Zhao et al., 2010; Chen et al., 2014) or clustering techniques (Su et al., 2008; Yu et al., 2011; Chen et al., 2016) to group potential aspect terms into different clusters (not explicit categories). Though this approach can automatically group opinion/aspect terms into different clusters or topics, it fails to explicitly classify a term into one of a set of user predefined categories.

For the task of aspect categorization, most existing methods assume the aspect terms be extracted in advance, and aim to predict their corresponding categories (Carenini et al., 2005; Yu et al., 2011; Zhai et al., 2010, 2011). To apply these methods to solve the problem studied in this paper, one needs to first use some aspect/opinion terms extraction method to identify aspect/opinion terms as a preprocessing step. In such a pipeline solution, error can be propagated across steps.

2.2 Deep Multi-task Learning

Multi-task learning aims to improve generalization for each individual task by exploiting relatedness among different tasks (Caruana, 1997). One common assumption in multi-task learning is that parameters for different tasks lie in a low-dimensional subspace (Argyriou et al., 2008; Kumar and III, 2012) which is achieved either by imposing low-rank constraint or matrix factorization. Through factorization, the model of each task becomes a linear combination of a small set of latent tasks. Following this idea, a multi-linear model was proposed in (Romera-Paredes et al., 2013) to deal with multi-modal tasks with multiple indexes. This tensor factorization idea also promotes a deep multi-task learning model (Yang et al., 2016) where the parameters in different layers of a CNN for different tasks form a tensor that could be factorized across tasks. Moreover, many deep learning models have been introduced for multi-task learning (Liu et al., 2015b; Misra et al., 2016) with an aim to learn shared hidden representation that are regularized from different tasks. Our proposed deep multi-task learning model is specially designed to suit in sentiment analysis and is expected to be more effective for the proposed finer-grained sentiment analysis compared with the general deep multi-task learning methods.

3 Preliminary

The base classifier used in our deep multi-task learning model is the coupled multi-layer attentions (CMLA) (Wang et al., 2017), which is proposed for aspect-opinion terms co-extraction. The basic component of CMLA is a pair of coupled attentions composing of an aspect attention and an opinion attention, which are interactively learned. The idea behind the coupled attentions is to exploit the relations among the aspect terms and opinion terms for double propagation (Qiu et al., 2011; Wang et al., 2016). Through the multi-layer design, both direct and indirect relations among terms can be captured.
Specifically, given a sentence with pre-trained word embeddings \( \{x_i\} \)'s, Gated Recurrent Unit (GRU) (Cho et al., 2014) is applied on top of \( x_i \) to obtain input feature representations \( \mathbf{H} = \{h_1, ..., h_{n_i}\} \). The architecture for the first layer of CMLA is shown in Figure 1, where an aspect prototype vector \( u^a \) and an opinion prototype vector \( u^p \) are first initialized for guiding the attention to select aspect and opinion terms. During learning, transformed hidden representations \( r^a_i \) and \( r^p_i \) are computed from the aspect attention \( f^a \) and the opinion attention \( f^p \), respectively, for each \( h_i \) through its interaction with two prototype vectors:

\[
\begin{align*}
    f^a(h_i, u^a, u^p) &= \tanh([h_i^\top G^a u^a : h_i^\top D^a u^p]), \\
    f^p(h_i, u^a, u^p) &= \tanh([h_i^\top G^p u^a : h_i^\top D^p u^p]),
\end{align*}
\]

where \([\cdot]\) denotes concatenation of vectors. \( G^a, G^p \in \mathbb{R}^{K \times d} \) and \( D^a, D^p \in \mathbb{R}^{K \times d} \) are 3-dimensional tensors composed of \( K \) bi-linear interaction matrices where each matrix models one type of implicit relation between \( h_i \) and \( u^a \) (or \( u^p \)). The core idea of coupled learning is reflected in the way that both aspect and opinion prototypes are used to compute each attention. Then \( r^a_i \) and \( r^p_i \) are obtained with GRU, given \( f^a(h_i, u^a, u^p) \) and \( f^p(h_i, u^a, u^p) \) as input, respectively, to incorporate context information. Let’s denote

\[
(r^a_i, r^p_i) = g(h_i, u^a, u^p; \mathbf{G}, \mathbf{D}, \mathbf{\theta}_{\text{GRU}}),
\]

where \( \mathbf{G} = \{G^a, G^p\}, \mathbf{D} = \{D^a, D^p\} \), and \( \mathbf{\theta}_{\text{GRU}} = \{\mathbf{\theta}^a_{\text{GRU}}, \mathbf{\theta}^p_{\text{GRU}}\} \) includes transformation matrices in GRU computation.

Additionally, an attention score \( e_i^a \) (or \( e_i^p \)) is computed to reflect the relevance of input \( h_i \) for aspect (or opinion) attention, and to compute sentence representation \( o^a \) (or \( o^p \)). We illustrate the computation for aspect attention as follows,

\[
    e_i^a = u^a \cdot r^a_i, \quad o^a = \sum_{i=1}^{n_i} \alpha_i^a h_i,
\]

where \( \alpha_i^a = \exp(e_i^a) / \sum_j \exp(e_j^a) \), and \( u^a \), which is to be learned, transforms vector \( r^a_i \) to attention score \( e_i^a \). Intuitively, \( o^a \) is dominated by the input feature vectors \( \{h_i\} \)'s with higher attention scores, which indicate more probable aspect/opinion terms. This will help to produce better prototype vector in the next layer, which in turn

\[u_{t+1}^a = \tanh(V^a \cdot u_t^a) + o_t^a,\]

where \( t \) is the index of a layer. In this way, \( u_{t+1}^a \) incorporates most probable aspect terms to select the other non-obvious target tokens in the next layer. The final prediction for aspects is made on the sum of hidden vectors \( r^a_i t \) for each layer \( t \), similar for opinions. The objective for training is to minimize the sum of cross-entropy losses of each token for both aspect prediction and opinion prediction.

### 4 Problem Statement and Motivation

Let \( C = \{1, 2, ..., C\} \) denote a predefined set of \( C \) categories, where \( c \in C \) is an entity/attribute type, e.g., “DRINK#QUALITY” in the restaurant domain. For any review sentence \( z_i = \{w_{i1}, ..., w_{in_i}\} \) consisting of a sequence of \( n_i \) tokens, we are given a collection of all the explicit aspect terms and opinion terms appearing in \( z_i \), as well as their corresponding categories, denoted by \( A_i = \{(a_{i1}, y_{i1}^a), ..., (a_{il_i}, y_{il_i}^a)\} \) for aspects, and by \( P_i = \{(p_{i1}, y_{i1}^p), ..., (p_{im}, y_{im}^p)\} \) for opinions, where \( y_{ij}^a, y_{ij}^p \subseteq C \). Our goal is to learn a predictive model \( f \) that takes \( z_i \) as input and generates \( A_i \) and \( P_i \) as outputs. Moreover, as an aspect or opinion term can be a phrase, we use the BIO encoding scheme to define labels. Specifically, for each category \( c \), we have five labels \( \{BA_c, IA_c, BP_c, IP_c, O_c\} \), where \( BA_c \) and \( IA_c \) refer to beginning of aspect and inside of aspect, respectively, for category \( c \), similar for \( BP_c \) and \( IP_c \) for opinions, and \( O_c \) refers to none of aspect or opinion terms for category \( c \).

To solve this problem using existing aspect/opinion terms extraction methods, e.g., CMLA, one straightforward solution is to train a CMLA model for each category \( c \), and combine the results of all CMLA models to generate final

---

\(^1\)Computation for opinion attention is similar
predictions. However, for each fine-grained category, aspect and opinion terms become extremely sparse, which makes it difficult to learn a precise model for each category if trained independently. To address this issue, we propose to model the problem in a multi-task learning manner, where term extraction for each category is considered as an individual task, and the goal is to jointly learn all the tasks by exploiting the commonalities and similarities among them. We hypothesize that some commonalities exist in the syntactic interactions among the tokens for different categories, and there are subtle relations among the categories that indicate their similarities to facilitate sharing. Moreover, global information on the overall categories of a sentence could be incorporated to assist token-level predictions.

These considerations are reflected in the following 3 components: 1) **Coupled Attentions with Shared Tensors**, which aims to model the commonalities in syntactic relations by sharing the tensor parameters \( \{ G, D \} \) among CMLA models, 2) **Context-aware Feature Sharing**, which aims to share features among tasks through constructing context-aware task similarity matrices, and 3) **Auxiliary Task**, which creates an auxiliary task to predict overall sentence-level category labels for helping token-level prediction tasks.

In the sequel, we name our proposed model as Multi-task Coupled Attentions (MTCA), whose overall architecture is shown in Figure 2. Specifically, MTCA takes the learning of CMLA for each category as a task \( T_c \), and a sequence of word embeddings \( \{ z_i \} \)’s as the input. Through the three components, each \( T_c \) produces a task-specific feature representation \( \tilde{r}_i^c \) for each word, as well as a task-specific context representation \( \tilde{o}_c \) for the sequence (i.e., sentence). Backpropagation is conducted on both the token-level prediction tasks and the auxiliary sentence-level prediction task to update the feature representations. In the following section, we present the 3 components in detail.

## 5 Proposed Methodology

### 5.1 Coupled Attentions with Shared Tensors

Similar to Section 3, for each task \( T_c \), we generate a pair of embeddings \( u_c = \{ u_c^a, u_c^p \} \)\(^2\) to capture the distributed representations of aspect prototype and opinion prototype, respectively, for category

\[ r_i^c = (r_i^{c,a}, r_i^{c,p}) = g(h_i, u_c^a, u_c^p, G_c, D_c, \theta_{GRU}) \]

for each task \( T_c \). Here different \( T_c \) corresponds to different tensors \( \{ G_c, D_c \} \) where \( G_c = \{ G^a_c, G^p_c \} \) and \( D_c = \{ D^a_c, D^p_c \} \) model the complex token interactions. We assume that these relations are similar across categories, and propose to learn a low-rank shared information among \( \{ G_c \} \)’s through tensor factorization. Specifically, let \( G^a \in \mathbb{R}^{C \times K \times d \times d} \) be the concatenation of all the \( \{ G^a_c \} \)’s and denote by \( G^k_a = G^a_{[k,:,:,:]} \in \mathbb{R}^{C \times d \times d} \) the collection of \( k \)-th bi-linear interaction matrices across \( C \) tasks for aspect attention. The same also applies for opinion attention. Factorization is performed on each \( G^a_k \) and \( G^p_k \), respectively, through

\[ G^a_{k[c,:]} = Z^a_{k[c,:]} G^a_k, \quad \text{and} \quad G^p_{k[c,:]} = Z^p_{k[c,:]} G^p_k, \] (6)

where \( G^a_k, G^p_k \in \mathbb{R}^{C' \times d \times d} \) are shared factors among all the tasks with \( C' < C \), while \( Z^a_k, Z^p_k \in \mathbb{R}^{C' \times C'} \) with each row \( Z^a_{k[c,:]} \) and \( Z^p_{k[c,:]} \) being specific factors for \( T_c \). The matrices of shared factors can be considered as \( C' \) latent basis interactions, where the original \( k \)-th bi-linear relation matrix \( G^a_{k[c,:]} \) (\( G^p_{k[c,:]} \)) for \( T_c \) is the linear combination of those latent basis interactions. In this way, we reduce the parameter dimensions by enforcing sharing within a small number of latent interactions. The same approach also applies to the tensors \( \{ D_c \} \)’s.

### 5.2 Context-aware Feature Sharing

Besides syntactic relations, we further explore similarities between tasks or categories to learn

---

\(^2\)We randomly initialize all category embeddings \( u_c \) from a uniform distribution: \( u_c \sim U[-0.2, 0.2] \in \mathbb{R}^d \).
more powerful features for different task. For example, “FOOD#PRICE” is more similar to “DRINK#PRICE” than “SERVICE#GENERAL” because the first two categories may share some common aspect/opinion terms, such as expensive. By representing each task or category in a form of distributed vector, we can directly compute their similarities to facilitate knowledge sharing. In this component, we aim to update features \( \tilde{r}_i^c \) from \( r_i^c \) by integrating task relatedness. To do that, we construct context-aware category representations which capture not only the general category information, but also the context it lies in. Follow (4), we obtain overall context representations \( o_c = \{ o_c^a, o_c^p \} \) for \( T_c \) through attention scores:
\[
\begin{align*}
o_c^a &= \sum_{i=1}^{n_i} \alpha_i^{(c,a)} h_i, & \text{and} & \quad o_c^p &= \sum_{i=1}^{n_i} \alpha_i^{(c,p)} h_i, \quad (7)
\end{align*}
\]
where \( \alpha_i^{(c,a)} \) and \( \alpha_i^{(c,p)} \) are normalized scores that represent the relevance of \( h_i \) for aspect and opinion attentions in \( T_c \), respectively. As a result, \( o_c^a \) and \( o_c^p \) will capture the features of the most likely aspect and opinion terms for category \( c \) within the sentence, as indicated by higher attention scores. Given \( o_c \), the context-aware category representation \( \tilde{u}_c = \{ \tilde{u}_c^a, \tilde{u}_c^p \} \) is generated through:
\[
\begin{align*}
\tilde{u}_c^a &= \tanh(\tilde{V}^a u_c^a) + o_c^a, & \tilde{u}_c^p &= \tanh(\tilde{V}^p u_c^p) + o_c^p
\end{align*}
\]
where \( \tilde{V}^a, \tilde{V}^p \in \mathbb{R}^{d \times d} \) are transformation matrices. Through this operation, \( \tilde{u}_c \) incorporates both general category features \( u_c \) and context-specific category features \( o_c \) for the input sentence. Let \( \tilde{U}^a, \tilde{U}^p \in \mathbb{R}^{C \times d} \) denote the matrices consisting of \( \tilde{u}_c^a, \tilde{u}_c^p \) as a row vector, respectively, the task similarity matrices in terms of aspects and opinions, \( S^a \) and \( S^p \), can be computed as follows,
\[
\begin{align*}
S^a &= q(U^a U^a \top), & S^p &= q(U^p U^p \top), \quad (8)
\end{align*}
\]
where \( q(\cdot) \) is the softmax function carried in a row-wise manner so that the similarity scores between \( T_c \) and any \( T_{c'} \) sum up to 1. The similarity matrices \( S^a \) and \( S^p \) are then used to refine feature representation for each task by incorporating feature representations from related tasks:
\[
\begin{align*}
\tilde{r}_i^{(c,a)} &= \sum_{c' = 1}^{C} S_{cc'}^a \tilde{r}_i^{(c',a)}, & \tilde{r}_i^{(c,p)} &= \sum_{c' = 1}^{C} S_{cc'}^p \tilde{r}_i^{(c',p)}, \\
\tilde{o}_c^a &= \sum_{c' = 1}^{C} S_{cc'}^a \tilde{o}_c^a, & \tilde{o}_c^p &= \sum_{c' = 1}^{C} S_{cc'}^p \tilde{o}_c^p,
\end{align*}
\]
We denote by \( \tilde{r}_i^c = \{ \tilde{r}_i^{(c,a)}, \tilde{r}_i^{(c,p)} \} \) the new feature representations for the \( i \)-th token and by \( \tilde{o}_c = \{ \tilde{o}_c^a, \tilde{o}_c^p \} \) the new feature representations for the overall context for the aspect attention and the opinion attention of task \( T_c \), respectively.

Note that the feature sharing among different tasks is context-aware because \( \tilde{U}^a \) and \( \tilde{U}^p \) are context-aware category representations. This means that different sentences might indicate different task similarities. For example, when cheap is presented, it might increase the similarities between “FOOD#PRICES” and “RESTAURANT#PRICES". As a result, \( \tilde{r}_i^{(c,a)} \) for \( T_c \) could incorporate more information from \( T_{c'} \) if \( T_{c'} \) has higher similarity score indicated by \( S_{cc'}^a \).

5.3 Multiple Layers
To capture non-obvious relations among aspect and opinion terms, we construct multiple layers of attentions. Given a sentence, each layer \( t \) produces its own \( \{ \tilde{r}_i^{(c,a)} \}_{c=1}^{C} \) for the \( i \)-th token, and \( \{ \tilde{o}_c \}_{c=1}^{C} \) for the whole sentence. Note that different layers takes the same \( H \) but different prototype vectors \( u_{c,t} \) as the input, where \( u_{c,t}^a \) (similar for \( u_{c,t}^p \)) is updated through:
\[
\begin{align*}
u_{c,t}^a &= \tanh(V^a u_{c,t-1}^a) + o_{c,t-1}^a. \quad (9)
\end{align*}
\]
The final feature representations are obtained as the sum of features from each layer:
\[
\begin{align*}
\gamma_i^{(c,a)} &= \sum_{t=1}^{T} \gamma_{i,t}^{(c,a)}, & \gamma_i^{(c,p)} &= \sum_{t=1}^{T} \gamma_{i,t}^{(c,p)} \quad (10) \\
\beta_c^a &= \sum_{t=1}^{T} \beta_{c,t}^a, & \beta_c^p &= \sum_{t=1}^{T} \beta_{c,t}^p, \quad (11)
\end{align*}
\]
where \( \gamma_i^{(c,a)} \) and \( \gamma_i^{(c,p)} \) are the final representations for \( i \)-th token, while \( \beta_c^a \) and \( \beta_c^p \) are the final representations for the input sentence for task \( T_c \).

5.4 Auxiliary Task
To better address the data sparsity issue, we aim to use additional global information on categories in the sentence level. Consider the following motivating example, if we know the sentence “The soup is served with nice portion, the service is prompt” has aspect/opinion terms from category “DRINKS#STYLE” and “SERVICE#GENERAL”, we can infer that some words in the sentence should belong to one of these two categories. To make use of this information, we
construct an auxiliary task to predict the categories of a sentence. Note that in training data, sentence-level labels can be obtained by integrating tokens’ labels. Therefore, besides the token loss for our target token-level prediction tasks, we also need to define sentence loss for the auxiliary task. The learning of the target task and auxiliary task are not independent. On one hand, the global sentence information helps the attention to select category-relevant tokens. On the other hand, if the attention are able to attend to target terms, the output context representation will filter out irrelevant noise, which helps the overall sentence prediction.

Denote by \( f_{\text{sen}}^c \) the classifier for the sentence-level prediction task for category \( c \), and by \( f_{\text{tok}} \) the classifier for task target, which are defined as,

\[
\hat{y}_c = f_{\text{sen}}^c (\beta^a_c, \beta^p_c) = q(W_c^a \beta^a_c + \beta^p_c),
\]

\[
\hat{y}_i^{(c,a)} = f_{\text{tok}} (\gamma_i^{(c,a)}; W_{a}^g) = q(W_{a}^g \gamma_i^{(c,a)}),
\]

\[
\hat{y}_i^{(c,p)} = f_{\text{tok}} (\gamma_i^{(c,p)}; W_{p}^g) = q(W_{p}^g \gamma_i^{(c,p)}),
\]

where \( q(\cdot) \) is the softmax function, \( \hat{y}_c \in \mathbb{R}^2 \) indicates the probability of the sentence belonging to category \( c \), and \( \hat{y}_i^{(c,a)}, \hat{y}_i^{(c,p)} \in \mathbb{R}^3 \) denoting the probabilities of being BA\(_{c}\) (BP\(_{c}\)), I\(_{a}\) (IP\(_{c}\)) or O\(_{c}\) for category \( c \). We use shared \( W^a \) (\( W^p \)) for token predictions because \( \gamma_i^{(c,a)} \) (\( \gamma_i^{(c,p)} \)) encodes the extent of interaction with prototype vectors that are not task-specific. The final label \( \hat{y}_i^c \) for \( i\)-th token for category \( c \) is produced by comparing the largest value in \( \hat{y}_i^{(c,a)} \) and \( \hat{y}_i^{(c,p)} \). If both of them are \( O \), then the final label is \( O \). If only one of them is \( O \), we pick the other one as the label. Otherwise, the final label is the one with the largest value. By incorporating the loss of the auxiliary task, the objective for our model can be written as:

\[
\min_{f_{\text{tok}}, f_{\text{sen}}} \sum_c \sum_{i} e(y_i, f_{\text{sen}}^c (\beta^a_c, \beta^p_c)) + \lambda \sum_c \sum_{i=1}^{n_i} \sum_{m \in \{a,p\}} e\left(y_i^{(c,m)} f_{\text{tok}} \left(\gamma_i^{(c,m)}; W_{m}\right)\right),
\]

where \( e(\cdot) \) is the cross-entropy loss, \( y_i \in \{0, 1\} \) indicates whether category \( c \) is presented for the input sentence, \( y_i^{(c,m)} \in \mathbb{R}^3 \) is a one-hot vector representing ground-truth label for \( i\)-th token, and \( \lambda \) is a trade-off parameter for the two types of losses.

### 5.5 Training

As shown in Figure 2, the training is carried by propagating the errors from top to bottom in an end-to-end manner. Specifically, we first obtain the loss from both the token-level prediction as well as the sentence-level prediction, which are backpropagated to \( \hat{r}_{i,t} \) and \( \bar{o}_{c,t} \) in each layer, respectively for each \( T_c \). These together will update \( S^a \), \( S^p \), which combines with the error from \( \bar{o}_{c,t} \) to update \( o_{c,t} \). Then the error from both \( \hat{r}_{i,t} \) and \( o_{c,t} \) will be used to update all the parameters for the coupled attentions until the word embeddings.

### 6 Experiments

The experiments are conducted on three benchmark datasets from subtask 1 in SemEval Challenge 2015 task 12 (Pontiki et al., 2015), SemEval Challenge 2016 task 5 (Pontiki et al., 2016), and SemEval Challenge 2014 task 4 (Pontiki et al., 2014), which are denoted by S1, S2, and S3, respectively. Note that S1 and S2 are reviews in restaurant domain and S3 is in laptop domain. We use term-level aspect-opinion annotations provided by (Wang et al., 2017) for S1 and S3, and manually annotate opinion terms for S2. To facilitate our experiment, we also make additional annotations on category labels for target terms, except for the aspect term categories for S1 and S2 which are provided by SemEval. The statistics of each dataset is shown in Table 1 where text and tuple represent the number of sentences and the number of tuples consisting of an aspect term and its corresponding category label, respectively. Each sentence may contain multiple aspect terms with more than one categories. The aspect categories are shown in Table 2. For S1 and S2, an aspect category is defined as the combination of an entity and an attribute, e.g., “FOOD#PRICES”. There are in total 12 categories. For S3, an aspect category is an entity.

We filter out some categories with a few target terms. The remaining categories are shown in Table 2.
Follow (Wang et al., 2017), we first obtain word embeddings by applying word2vec\(^4\) on Yelp Challenge dataset\(^5\) and electronic domain in Amazon reviews (McAuley et al., 2015) for restaurant and laptop datasets, respectively. We set the dimension of word embeddings to be 150 and the dimension after GRU transformation to be 50 for all the three datasets. We use two layers as in CMLA for experiments. For each layer, the number of bilinear interactions for the 3-dimensional tensors is 20 ($K=20$), and tensor factorization operates with $C'=5$ for S1 and S2, and $C'=8$ for S3. We apply partial dropout at 0.5 to chosen parameters to avoid overfitting. The training is carried with rmsprop with the initial value at 0.001 and decayed with rate 0.9. The trade-off parameter $\lambda$ is set to be 1.0. All the hyper-parameters are chosen according to cross-validation.

### 6.1 Experimental Results

We conduct comparison experiments with the following baseline models:

- **NLANG**: The best system for both SemEval-15 and SemEval-16 for the proposed task.
- **IHS-RD, XRCE**: The second best systems for SemEval-15 and SemEval-16, respectively.
- **RNCRF+**: We modify RNCRF (Wang et al., 2016), which is for aspect-opinion terms extraction, by defining finer-grained categories as labels.
- **CMLA+**: Similar to RNCRF+, we modify CMLA (Wang et al., 2017) by defining finer-grained categories as labels.
- **CMLA++**: CMLA is used to extract all the aspect and opinion terms, and then a category classification layer is added to classify the extracted terms.

We report the results from top performing systems in the Challenges for S1 and S2. There are no reported results for S3 as the original task is different from ours. Note that original task for S1 and S2 includes two slots: slot 1 for aspect terms extraction and slot 2 for aspect category prediction. Moreover, SemEval made the combination of slot 1 and slot 2 as an additional task that corresponds to the problem we study. However, most of the reported models did not provide feasible methods for the joint prediction of aspect terms and corresponding categories, instead, they trained the model for slot 1 first and then combined with slot 2. This may fail to capture the relations between target terms and their categories. In order to show the advantage of our model, we modify the existing state-of-the-art deep models for aspect/opinion terms extraction to suit our problem settings. Since RNCRF and CMLA both exploit the correlations between aspect terms and opinion terms, which have been shown to be effective for extraction task, a simple idea is to increase the number of classes to incorporate different categories, e.g., BA becomes $\{BA_c\}$’s for different category $c$. By increasing the number of classes, the only change to the original model is the dimension of classification matrix. As a result, the modified model should be able to capture the correlations between target terms based on their categories. On the other hand, we also construct another baseline model (denoted by CMLA++) based on CMLA by separating the task into 2 steps. The first step is the same as CMLA for extracting target terms. Then the second step performs category prediction only on the extracted terms.

The comparison results are shown in Table 3. It can be seen that MTCA achieves the state-of-the-art performances in category-specific aspect and opinion terms extraction (ASC and OPC). And there is a large gap between the results of MTCA and the other baseline models on S1 and S2. This is because RNCRF+ and CMLA+ can only propagate information between target terms within each category, but fail to explore the relations and commonalities among different categories. The other model CMLA++ performs even poorer, because the training is separated into different stages, similar to the top systems in SemEval Challenges. This separation results in the failure of propagating information from category prediction to target term extraction. The result proves the effectiveness of MTCA for learning shared information among different tasks, as well as the addition of global information to assist extraction. The improvement for S3 is not significant, which might indicate that the category correlations are not obvious in laptop domain, as can be seen in Table 2. Only entity labels make different categories distinct from each other.

Moreover, we also report the results on target terms extraction (AS and OP) by accumulating the aspect/opinion terms that are assigned at least one category by MTCA. It can be seen that MTCA still achieves comparable performances even if the data becomes sparser when adding the category

---

\(^4\)https://radimrehurek.com/gensim/models/word2vec.html

\(^5\)http://www.yelp.com/dataset\_challenge
Table 3: Comparison results in terms of $F_1$ scores. ASC (OPC) refers to category-specific aspect (opinion) terms extraction. AS (OS) refers to aspect (opinion) terms extraction.

| Model   | ASC   | OPC   | AS   | OP   | ASC   | OPC   | AS   | OP   |
|---------|-------|-------|------|------|-------|-------|------|------|
| NLANG   | 42.90 | -     | 67.11| -    | 52.61 | -     | 72.34| -    |
| HSIRD   | 42.72 | -     | 63.12| -    | -     | -     | -    | -    |
| XRCE    | 48.89 | -     | 61.98| -    | -     | -     | -    | -    |
| RNCRF+  | 54.00 | 47.86 | 67.74| 67.62| 56.04 | 51.09 | 69.74| 74.26|
| CMLA+   | 57.35 | 55.70 | 70.73| 73.68| 57.83 | 56.04 | 75.21| 77.90|
| CMLA++  | 53.46 | 53.94 | 70.73| 73.68| 54.05 | 54.34 | 75.21| 77.90|
| MTCA    | 63.16 | 59.17 | 71.31| 72.23| 65.34 | 61.44 | 73.26| 76.10|

Table 4: Comparison results with reductions.

| Different Components | S1         | S2         |
|----------------------|------------|------------|
| ASC (OPC)            | ASC (OPC)  |            |
| MTCA (C1+C2+C3)      | 63.16      | 59.17      |
| C1+C3                | 61.95      | 58.57      |
| C2+C3                | 61.67      | 55.89      |
| C2+C3*               | 61.30      | 55.30      |
| C1+C2                | 60.67      | 56.97      |
| C3                   | 60.18      | 57.03      |

Figure 3: Sensitivity studies for data S1.

Figure 4: Examples of learned attention scores.

information. On the contrary, the results for RNCRF+, CMLA+ and CMLA++ are obtained using the original models that ignore category labels.

As have been discussed in the previous sections, the multi-task attention network explores the commonalities and relations among tasks through both tensor sharing and feature sharing, as well as enhances prediction results by incorporating auxiliary labels. To test the effect of each component, we conduct comparison experiments for different combinations of these components as shown in Table 4, where C1, C2 and C3 represents separate component for multi-task tensor sharing, context-aware feature sharing and auxiliary task, respectively. Note that C2+C3* refers to the use of same tensor across all the tasks. Clearly, The inclusion of either feature sharing (C2+C3) or tensor sharing (C1+C3) improves the results compared to independent training (C3). Furthermore, tensor sharing proves to be more beneficial than feature sharing. This shows that the commonalities in terms of token interactions are more obvious for different tasks. Moreover, either independent tensors (C2+C3) or the same tensor (C2+C3*) across tasks does not perform well. This indicates that it is still crucial to explore both the uniqueness and commonality of all the tasks, which are preserved in our proposed model. By comparing the results between C1+C2+C3 and C1+C2, we can see how auxiliary task contributes to the final prediction (2.49% and 4.05% increase for ASC in S1 and S2 respectively). This shows that the global information in the sentence level could enforce the correct predictions of each token within the sentence by reiterating the category information.

To show the robustness of our model, we test it with different dimension of factorization ($C'$). The results on S1 are shown in Figure 3. We also provide some examples in Figure 4 to show what the attentions learn for different categories. The first column shows the largest attention scores for both aspects and opinions in each identified category. We use different colors to denote different categories and list the identified categories in the second column. Clearly, MTCA is able to attend important tokens for different categories for extraction. Moreover, MTCA is able to identify the case when specific terms belong to more than one categories, which are indicated with multiple colors in the figure.

7 Conclusion

In this work, we introduce a finer-grained task involving the predictions of both aspect/opinion terms and their corresponding aspect categories, and offer a novel multi-task deep learning model, MTCA, to solve the problem. The model is able to exploit syntactic commonalities and task simi-
larities through attention mechanism. In the end, we demonstrate the effectiveness of our model on three benchmark datasets.

References
Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. Mach. Learn. 73(3):243–272.

Giuseppe Carenini, Raymond T. Ng, and Ed Zwart. 2005. K-cap. pages 11–18.

Rich Caruana. 1997. Multitask learning. Mach. Learn. 28(1):41–75.

Lu Chen, Justin Martineau, Doreen Cheng, and Amit P. Sheth. 2016. Clustering for simultaneous extraction of aspects and features from reviews. In NAACL-HLT. pages 789–799.

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In ACL. pages 347–358.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In EMNLP. pages 1724–1734.

Honglei Guo, Huijia Zhu, Zhi Li Guo, Xiaoxun Zhang, and Zhong Su. 2009. Product feature categorization with multilevel latent semantic association. In CIKM. pages 1087–1096.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In KDD. pages 168–177.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In EMNLP. pages 1035–1045.

Wei Jin and Hung Hay Ho. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In ICML. pages 465–472.

Abhishek Kumar and Hal Daumé III. 2012. Learning task grouping and overlap in multi-task learning. In ICML.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In COLING. pages 653–661.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015a. Fine-grained opinion mining with recurrent neural networks and word embeddings. In EMNLP. pages 1433–1443.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015b. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In NAACL. pages 912–921.

Yue Lu, ChengXiang Zhai, and Neel Sundaesaran. 2009. Rated aspect summarization of short comments. In WWW. pages 131–140.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In SIGIR. pages 43–52.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphe De Clercq, Vronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Louchakvitchev, Evgeny Kotelnikov, Nuria Bel, Salud Maria Jimnez-Zafra, and Glen Eryiit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In SemEval 2016.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In 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 SemEval. pages 27–35.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In EMNLP. pages 339–346.

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

Bernardino Romera-Paredes, Hane Aung, Nadia Bianchi-Berthouze, and Massimiliano Pontil. 2013. Multilinear multitask learning. In ICML (3). volume 28 of JMLR Workshop and Conference Proceedings. pages 1444–1452.

Qi Su, Xinying Xu, Honglei Guo, Zhi Li Guo, Xian Wu, Xiaoxun Zhang, Bin Swen, and Zhong Su. 2008. Hidden sentiment association in chinese web opinion mining. In WWW. pages 959–968.

Ivan Titov and Ryan McDonald. 2008a. Modeling online reviews with multi-grain topic models. In www. pages 111–120.

Ivan Titov and Ryan T. McDonald. 2008b. A joint model of text and aspect ratings for sentiment summarization. In ACL. pages 308–316.
Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In EMNLP.

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

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In EMNLP. pages 1533–1541.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In NAACL. pages 1480–1489.

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

Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. 2011. Aspect ranking: Identifying important product aspects from online consumer reviews. In ACL.

Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2010. Grouping product features using semi-supervised learning with soft-constraints. In COLING. pages 1272–1280.

Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2011. Clustering product features for opinion mining. In WSDM. pages 347–354.

Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In EMNLP. pages 56–65.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In CIKM. pages 43–50.