Bridging Towers of Multi-task Learning with a Gating Mechanism for Aspect-based Sentiment Analysis and Sequential Metaphor Identification

Rui Mao, Xiao Li
Ruimao Tech, Shenzhen, China
{mao.r, li.x}@ruimao.tech

Abstract

Multi-task learning (MTL) has been widely applied in Natural Language Processing. A major task and its associated auxiliary tasks share the same encoder; hence, an MTL encoder can learn the sharing abstract information between the major and auxiliary tasks. Task-specific towers are then employed upon the sharing encoder to learn task-specific information. Previous works demonstrated that exchanging information between task-specific towers yielded extra gains. This is known as soft-parameter sharing MTL. In this paper, we propose a novel gating mechanism for the bridging of MTL towers. Our method is evaluated based on aspect-based sentiment analysis and sequential metaphor identification tasks. The experiments demonstrate that our method can yield better performance than the baselines on both tasks. Based on the same Transformer backbone, we compare our gating mechanism with other information transformation mechanisms, e.g., cross-stitch, attention and vanilla gating. The experiments show that our method also surpasses these baselines.

Introduction

Parameter-sharing-based multi-task learning (MTL) has been widely applied in diverse Natural Language Processing (NLP) tasks (Ruder et al. 2019; Dankers et al. 2019; Chen and Qian 2020). Given a set of related tasks, MTL aims to use a unified model to learn their sharing representations (Zhang and Yang 2017). Compared to individual processing of a single task, introducing related auxiliary tasks can further boost the performance of a Deep Neural Network (DNN)-based MTL model on the major task. This is because multiple related tasks support the sharing encoder of the MTL model in acquiring knowledge from each subtask (Zhang and Yang 2017). Thus, the major task is improved by applying the learnt sharing knowledge from different tasks. Besides, introducing multi-tasks can reduce the risk of overfitting in DNNs (Ruder 2017). Based on the sharing encoder, an MTL model has task-specific towers (a tower refers to a stack of DNN layers) to learn task-specific knowledge for each subtask. Previous works (Zhang and Yang 2017; Ruder 2017) demonstrated that exchanging information between towers can further improve MTL performance, which is known as soft-parameter sharing. In this paper, we propose a novel information transformation (soft-parameter sharing) mechanism between MTL towers, namely Gated Bridging Mechanism (GBM).

The motivation of introducing GBM is that gating mechanism is intuitive for the filtering of information from auxiliary tasks to a main task, because the main task tower can decide through gating how much information is utilised from its private tower and its neighbour towers (the auxiliary task towers). Compared with previous information transformation methods, e.g., cross-stitch (Misra et al. 2016), attention (Chen and Qian 2020), and vanilla gating (Dankers et al. 2019), whose mechanisms fuse the information from private and neighbour towers directly, GBM takes an extra gate to filter out useless information from neighbour towers before fusion. The output of GBM is given by the trade-off between the information from a private tower and the filtered information from neighbour towers. Such a method allows an MTL model to absorb selected information from auxiliary tasks, hence yielding better performance.

We examine our method with two independent MTL tasks, namely Aspect-based Sentiment Analysis (ABSA) and Sequential Metaphor Identification (SMI). ABSA aims to identify aspect terms, opinion terms and sentiment polarities of the aspect terms in a sequence. We introduce ABSA in our tests, because conventionally, ABSA has multiple predefined subtasks which are related to each other (He et al. 2019; Chen and Qian 2020). SMI, on the other hand, aims to identify the metaphoricity of each token in a sequence. We introduce SMI because it can use a single-task sequence labelling model (Wu et al. 2018; Gao et al. 2018) or an MTL model with an auxiliary task (Dankers et al. 2019; Chen et al. 2020). Thus, we can employ different auxiliary tasks with different utilities for SMI to examine, if GBM can gain information from a supportive auxiliary task, e.g., Part-of-Speech tagging, or filter out useless information which is given by randomly generated hidden states.

By testing with three publicly available datasets developed by Pontiki et al. (2014) and Steen et al. (2010) for ABSA and SMI, respectively, our method yields an average gain of 1.11% against the strongest baselines on both tasks. Furthermore, we compare GBM against cross-stitch, attention and vanilla gating mechanisms. The experiments show that GBM outperforms these methods based on the same backbone of BERT (Devlin et al. 2019) and Transformer

(Vaswani et al. 2017). Finally, we experimentally demonstrate that GBM can functionally filter out useless information and gain useful information from an auxiliary task.

The contribution of this work can be summarised as twofolds: (1) We propose a novel Gated Bridging Mechanism (GBM) between multi-task learning (MTL) towers, which outperforms previous information transformation methods. (2) Our GBM-based MTL model achieves better performance than previous baselines on aspect-based sentiment analysis and sequential metaphor identification tasks.

Related Work

Multi-task Learning

parameter sharing-based MTL takes advantages in modelling low resource data (Duong et al. 2015), learning from different tasks (Zhang and Yang 2017) and reducing overfitting risks (Ruder 2017). It can be categorised as hard-parameter sharing methods, soft-parameter sharing methods, and their combinations.

According to Guo, Lee, and Ulbricht (2020), for hard-parameter sharing methods, all tasks share the same backbone parameters; alternatively, part of backbone parameters are shared, while each task has its private task-specific tower upon the sharing backbone (Dong et al. 2015; Long et al. 2017). For soft-parameter sharing, each task has its private tower, while the activated private parameters are shared or constrained by different mechanisms between tasks (Duong et al. 2015; Liu, Qiu, and Huang 2016). Recently, the combinations of hard-parameter sharing and soft-parameter sharing methods are widely applied. Apart from different parameter regularisation (Duong et al. 2015; Ruder et al. 2017) and loss functions (Cipolla, Gal, and Kendall 2018), we are more interested in information transformation mechanisms. Misra et al. (2016) proposed Cross-stitch Networks, where information from previous layers in different towers was linearly combined, passing to current layers. Liu, Qiu, and Huang (2016) proposed Recurrent MTL Networks, where the backbone of a subtask was based on LSTM (Hochreiter and Schmidhuber 1997). The information between different towers was exchanged by using a vanilla gating mechanism. Attention was another widely applied mechanism for fusing information from a main task and its auxiliary tasks, although there were slight modifications on attention for fitting to the tasks (Liu, Johns, and Davison 2019; Liu et al. 2019b; He et al. 2019; Chen and Qian 2020).

Compared with these information transformation methods, our proposed GBM can selectively reject useless information from an auxiliary task, if the auxiliary task is not supportive for the major task. For more details, please view the sections of Comparison between Different Bridges and Gated Bridging Mechanism Analysis.

Aspect-based Sentiment Analysis

Two types of methods were commonly employed in ABSA, namely separate methods and unified methods. For separate methods, previous works adopted a pipeline style (Hu et al. 2019), extracting aspect terms first (Wang et al. 2017; Xu et al. 2018), then identifying sentiment polarities (Li et al. 2018; Chen and Qian 2019) for the aspect terms. Recently, unified MTL models (Li et al. 2019a; Luo et al. 2019; He et al. 2019) are more popular in ABSA, because a sentiment classification task can benefit from learning the related subtasks (Chen and Qian 2020), e.g., aspect extraction and opinion extraction, yielding better results. He et al. (2019) processed ABSA, document-level sentiment classification and document-level domain classification tasks simultaneously with MTL. They employed a fully-connected layer with ReLU activation as the information transformation mechanism, passing the concatenated hidden states of subtasks from a previous training iteration to the current iteration. Chen and Qian (2020) proposed a unified ABSA model, by modelling the relations between different subtasks with an attention-based information transformation mechanism. Their model achieved state-of-the-art (SOTA) performance in ABSA. Conventional supervised learning-based ABSA is domain-dependent, thus Li et al. (2019b); Zheng et al. (2020) used transfer learning to address the challenge of cross-domain ABSA.

Sequential Metaphor Identification

Identifying metaphors is a widely studied semantic task that was modelled with different learning paradigms (Shutova, Kiela, and Maillard 2016; Mao, Lin, and Guerin 2018; Le, Thai, and Nguyen 2020; Su et al. 2020). SMI is a token level sequence tagging learning task, thus single task learning was commonly used (Wu et al. 2018; Mao, Lin, and Guerin 2019). Currently, there are more MTL models applied in SMI with different auxiliary tasks, outperforming previous single task learning models. Le, Thai, and Nguyen (2020) proposed a Graph Convolutional Neural Network with dependency tree-based MTL. They introduced an auxiliary task of Word Sense Disambiguation to support the main task prediction, where the main task and auxiliary task were trained alternatively. Chen et al. (2020) employed hard-parameter sharing method with BERT as the backbone, and fully-connected layers as task-specific towers, where the auxiliary task is idiom prediction. Dankers et al. (2019) employed MTL models with different emotional auxiliary tasks for SMI, e.g., predicting numerical scores of valence, arousal and dominance. They examined different soft-parameter sharing mechanisms, e.g., cross-stitch (Misra et al. 2016) and vanilla gating (Liu, Qiu, and Huang 2016), while their BERT and Bi-LSTM (Graves and Schmidhuber 2005) based hard-parameter sharing method with an auxiliary valence prediction yielded the best result.

Wu et al. (2018); Gong et al. (2020); Su et al. (2020) demonstrated that learning Part-of-Speech (PoS) features can boost SMI learning performance. Thus, we introduce PoS tagging as an auxiliary task in our MTL model to differentiate metaphorical and literal senses in different PoS.

Methodology

Task Definition

ABSA ABSA has three conventional subtasks, namely aspect extraction (AE), opinion extraction (OE) and sentiment classification (SC). Each subtask is considered as a se-

(a)	Input:	Even	though	it's	good	seafood	,	the	prices	are	too	high	.
	AE:	O	O	O	O	B	O	O	B	O	O	O	O
	OE:	O	O	O	B	O	O	O	O	O	O	B	O
	SC:	neu	neu	neu	neu	pos	neu	neu	neg	neu	neu	neu	neu
(b)	Input:	I	have	already	passed	the	two	written	exams	.			
	SMI:	lit	lit	lit	met	lit	lit	lit	lit	lit			
	PoS:	PRON	AUX	ADV	VERB	DET	NUM	VERB	NOUN	PUNCT			

Table 1: Example labels for (a) aspect-based sentiment analysis; (b) sequential metaphor identification.

quence tagging task, which is in line with Chen and Qian (2020). Given an input sequence with a length of L tokens (i.e. $t_1, t_2, ..., t_L$), an MTL model aims to predict three label sequences which indicate aspect terms, opinion terms and sentiment polarities of the aspect terms, respectively. Following Chen and Qian (2020), we employ the BIO annotation paradigm for AE and OE learning, where BIO defines the beginning, the inside or the outside of a target label. Given $t_1, ..., t_L$, the annotated labels of AE are $Y^{AE} = y_1^{AE}, ..., y_L^{AE}$, and the labels of OE are $Y^{OE} = y_1^{OE}, ..., y_L^{OE}$, where $y_k^{AE}, y_k^{OE} \in \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}$. The associated SC labels are $Y^{SC} = y_1^{SC}, ..., y_L^{SC}$, where $y_k^{SC} \in \{\mathrm{pos}, \mathrm{neg}, \mathrm{neu}\}$, indicating positive, negative and neutral sentiment polarities, respectively. An example of ABSA can be viewed in Table 1a.

SMI We aim to predict a sequence of labels that indicates the metaphoricity of each token in SMI. Similarly, given $t_1, ..., t_L, Y^{SMI} = y_1^{SMI}, ..., y_L^{SMI}$, where $y_k^{SMI} \in \{\mathrm{met}, \mathrm{lit}\}$ with met and lit representing metaphoric and literal, respectively. PoS labels of input texts are automatically generated by spaCy toolkit (Honnibal and Montani 2017), following the Universal Dependencies scheme[1]. An example of SMI can be viewed in Table 1b.

Framework

Figure 1 shows the overall framework of our mode. First, an input sequence $(t_1, ..., t_L)$ is encoded with the backbone of BERT, yielding the sharing hidden states (H^s) by

$$H^s = BERT(t_1, ..., t_L). \quad (1)$$

Then, we use multiple Transformer layers as a task-specific tower. In ABSA, there are three subtasks (τ_1 = AE, τ_2 = OE, τ_3 = SC). Thus three task-specific towers are employed upon BERT. In SMI, there are two subtasks (τ_1 = SMI, τ_2 = POS). Hence, two task-specific towers are employed upon BERT. Following the first Transformer layer, there is a stack of blocks in each tower. Each block, e.g., Block i, where $i \in \{1, 2, ..., n\}$ consists of a Gated Bridging Mechanism layer (GBM) and a Transformer layer (see Figure 1). The first Transformer layer in a task-specific tower is defined as Block 0. We use $G_i^{\tau_j}$ to denote the output of the GBM in Block i of the task τ_j tower. Noticeably, a GBM in Block i uses the Transformer hidden states of its previous block in each tower ($H_{i-1}^{\tau_j}$) as input

$$G_i^{\tau_j} = GBM_{\phi_{i,j}}(H_{i-1}^{\tau_1}, ..., H_{i-1}^{\tau_j}, ...), \quad (2)$$

[1]We also tested Penn Treebank PoS annotation paradigm (Marcus, Santorini, and Marcinkiewicz 1993), while it did not yield better performance on our dataset.

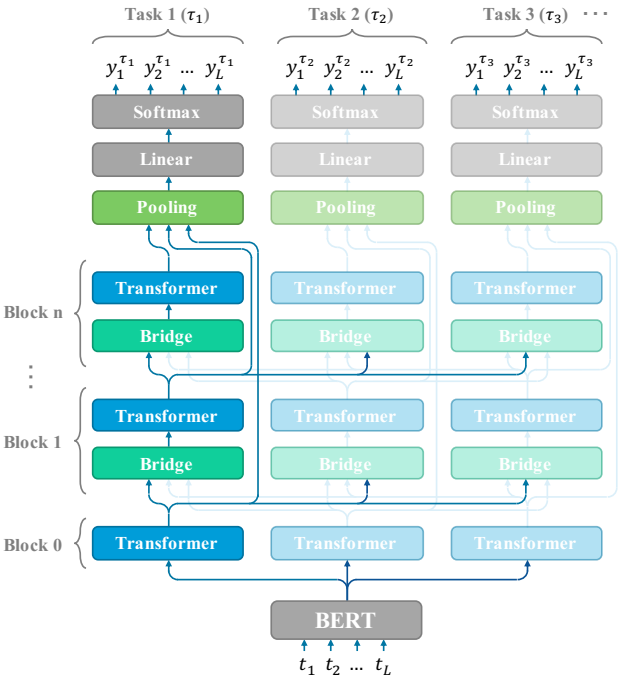

Figure 1: The framework of Multi-task learning with Gated Bridging Mechanism. There are three subtasks (τ_1, τ_2, τ_3) in the model. Each subtask consists of $n + 1$ blocks.

where ϕ denotes learnt parameters in GBM. The Transformer hidden states in each block are given by

$$\begin{cases} H_0^{\tau_j} = Trans_{\phi_{0,j}}(H^s), \\ H_i^{\tau_j} = Trans_{\phi_{i,j}}(G_i^j), \quad 0 < i \leq n. \end{cases} \quad (3)$$

According to Peters et al. (2018b) and Liu et al. (2019a), different Transformer layers in a Language Model have different utilities in semantic and syntactic down-stream tasks. Our hypothesis is that a weighted sum pooling strategy will ensure the best use of features from each Transformer layer in a task-specific tower. Thus, the pooling features after the last block in a tower are given by

$$H_{pool}^{\tau_j} = \sum_{i=0}^{n} \alpha_i^{\tau_j} H_i^{\tau_j}, \quad (4)$$

where $\alpha_i^{\tau_j} \in \mathbb{R}$ is a learnt parameter.

Finally, the output of Task τ_j Tower is given by a linear

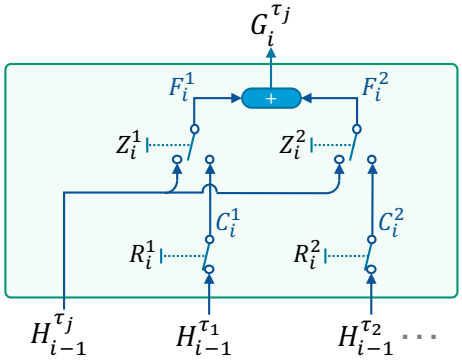

Figure 2: An illustration of the proposed Gated Bridging Mechanism with three subtask examples, where i denotes Block i. H is Transformer hidden states. $\{\tau_j, \tau_1, \tau_2, ...\}$ are different subtasks, where τ_j is the focused task in a multi-task learning tower; R is a reset gate; C is new current states; Z is an update gate; F is the fusion of H^{τ_j} and C. '+' is the element-wise addition. G is the output of GBM.

projection:

$$\hat{Y}^{\tau_j} = W_{fc}^{\tau_j} H_{pool}^{\tau_j} + b_{fc}^{\tau_j}, \quad (5)$$

where $W_{fc}^{\tau_j}$ and $b_{fc}^{\tau_j}$ are learnt parameters. We employ a cross-entropy loss function for each subtask, where the cross-entropy criterion integrates the final softmax function in Figure 1. The MTL model loss ($\mathcal{L}$) is given by the weighted (ω) sum of subtask losses

$$\mathcal{L} = \sum_{\tau_j} \omega^{\tau_j} \text{CrossEntropy}(\hat{Y}^{\tau_j}, Y^{\tau_j}), \quad (6)$$

where ω^{τ_j} is a hyper-parameter.

Gated Bridging Mechanism

$GBM_{\phi_{i,j}}(\cdot)$ in E.q. 2 is the GBM operation in Block i of task τ_j. It uses Transformer hidden states $(H_{i-1}^{\tau_1}, ..., H_{i-1}^{\tau_j}, ...)$ from the previous blocks from all subtasks as inputs and generates an output $G_i^{\tau_j}$. We explain $GBM_{\phi_{i,j}}(\cdot)$ by employing relevant gating concepts from Cho et al. (2014). Figure 2 shows the graphic depiction of $GBM_{\phi_{i,j}}(\cdot)$.

We use H^{τ_j} to denote that Transformer hidden states are from the tower of a focused task τ_j. H^{τ_m} $(m \neq j)$ denotes the hidden states that are derived from one of the neighbour towers of τ_j. First, we employ reset gates in τ_j to filter the corresponding hidden states from neighbour towers. E.g., a reset gate R_i^m is given by

$$R_i^m = \sigma(W_{\phi_{R,i,j}}^m H_{i-1}^{\tau_m} + b_{\phi_{R,i,j}}^m), \quad (7)$$

where σ denotes the sigmoid activation function.

Then, we apply the reset gate R_i^m to filter $H_{i-1}^{\tau_m}$, hence, generating new current states C_i^m by

$$C_i^m = tanh(W_{\phi_{C,i,j}}^m (R_i^m \odot H_{i-1}^{\tau_m}) + b_{\phi_{C,i,j}}^m), \quad (8)$$

Dataset	Data	Train	Valid	Test
Laptop	# of seq.	2,439	609	800
	# of tok.	38,675	9,670	11,007
	% of ap.	7.3	7.3	9.8
	% of op.	5.9	6.1	6.6
	% of pos.	2.9	2.6	4.5
	% of neg.	2.7	2.9	1.9
	% of neu.	1.6	1.7	3.1
Restaurant	# of seq.	2,436	608	800
	# of tok.	35,545	8,779	11,825
	% of ap.	11.8	11.0	14.0
	% of op.	8.2	8.4	9.2
	% of pos.	7.2	6.8	9.2
	% of neg.	2.3	2.2	2.1
	% of neu.	2.1	1.7	2.6
VUA	# of seq.	6,323	1,550	2,694
	# of tok.	116,622	38,628	50,175
	% of met.	11.2	11.6	12.4
	% of lit.	88.8	88.4	87.6

Table 2: Data statistics. # denotes the number; seq denotes sequences; % denotes the percentage of tokens (tok) with a specific label among all tokens; ap denotes aspect labels; op denotes opinion labels; pos denotes positive aspects, neg denotes negative aspects; neu denotes neutral aspects; met denotes metaphors; lit denotes literals.

where $\odot$ denotes element-wise product. R^m is functioned by controlling the number of activated neurons. If large parts of the neurons are close to 0, much of the information in $H_{i-1}^{\tau_m}$ is filtered by R_i^m. In Figure 2, e.g., R_i^1 controls whether $H_{i-1}^{\tau_1}$ flows into the next fusion step. Besides, we employ a non-linear projection function on the selected $H_{i-1}^{\tau_m}$, because we believe that $H_{i-1}^{\tau_m}$ and $H_{i-1}^{\tau_j}$ are in different vector spaces. The non-linear operation can project the selected $H_{i-1}^{\tau_m}$ to the space of $H_{i-1}^{\tau_j}$. Thus, C_i^m and $H_{i-1}^{\tau_j}$ can be added in the following operation.

Next, we employ an update gate Z_i^m to control if $H_{i-1}^{\tau_j}$ should fuse with C_i^m. Z_i^m is given by

$$Z_i^m = \sigma(W_{\phi_{Z,i,j}}^m H_{i-1}^{\tau_j} + b_{\phi_{Z,i,j}}^m + V_{\phi_{Z,i,j}}^m C_i^m + d_{\phi_{Z,i,j}}^m), \quad (9)$$

where V and d are learnt parameters. The fused features F_i^m are given by the trade-off between $H_{i-1}^{\tau_j}$ and C_i^m, where the trade-off is controlled by Z_i^m

$$F_i^m = Z_i^m \odot H_{i-1}^{\tau_j} + (1 - Z_i^m) \odot C_i^m. \quad (10)$$

As seen in Figure 2, the more information F_i^1 fuses from $H_{i-1}^{\tau_j}$, the more information from C_i^1 is filtered by Z_i^1.

Finally, the output ($G_i^{\tau_j}$) of $GBM_{\phi_{i,j}}(\cdot)$ is activated by

$$G_i^{\tau_j} = \sigma(W_{\phi_{G,i,j}}^{\tau_j}(\sum_{m \neq j} F_i^m) + b_{\phi_{G,i,j}}^{\tau_j}). \quad (11)$$

Experiments

Datasets

We employ a laptop dataset and a restaurant dataset from Pontiki et al. (2014) for ABSA, and the largest all word annotated metaphor dataset, VU Amsterdam Metaphor Corpora (VUA) from Steen et al. (2010) for SMI. These are

Model	Laptop				Restaurant			
	AE-F1	OE-F1	SC-F1	ABSA-F1	AE-F1	OE-F1	SC-F1	ABSA-F1
He et al. (2019)-GloVe	78.46	78.14	69.92	57.66	84.01	85.64	71.90	68.32
He et al. (2019)-BERT$^\sharp$	77.55	81.00	75.56	61.73	84.06	85.10	75.67	70.72
Hu et al. (2019)-BERT$^\sharp$	82.34	-	62.50	61.25	86.71	-	71.75	73.68
Chen and Qian (2020)-BERT	81.79	79.72	73.91	63.40	86.38	87.18	81.61	75.42
GBM-MTL-BERT-ours	**83.34***	77.93	**77.52***	**65.61***	**87.10***	87.16	**82.24***	**75.73***

Table 3: Model performance on aspect-based sentiment analysis. AE is aspect extraction. OE is opinion extraction. SC is sentiment classification. ABSA-F1 is an overall measure for AE and SC. * denotes the improvement is statistically significant based on a 2-tailed test ($p < 0.05$). Underline denotes the best baseline performance. $^\sharp$ was reported by Chen and Qian (2020).

widely applied benchmark datasets on both tasks (Hu et al. 2019; He et al. 2019; Chen and Qian 2020; Dankers et al. 2019; Mao, Lin, and Guerin 2019; Le, Thai, and Nguyen 2020). Since training and testing sets of the laptop and restaurant datasets had fixed segmentation in SemEval-2014 Task 4: Aspect-based Sentiment Analysis (Pontiki et al. 2014), following Chen and Qian (2020), we use their randomly selected 20% of samples from the training sets as validation sets for fine-tuning hyper-parameters of our model. The rest of 80% samples are used for training. For the VUA dataset, we employ the training, validation and testing sets that were firstly developed by Gao et al. (2018). Relevant statistics can be viewed in Table 2.

Baselines

He et al. (2019) proposed a unified MTL model for learning AE-OE, SC, document-level sentiment classification and domain classification, where the AE-OE prediction is a unified subtask, based on GloVe (Pennington, Socher, and Manning 2014). For a fair comparison, we compare our model with their BERT-based AE-OE and SC MTL model that was reported by Chen and Qian (2020).

Hu et al. (2019) proposed a pipeline style ABSA method. They modelled AE first. AE span representation in the AE model is then fed to a SC classifier for enhancing sentiment polarity predictions. We benchmark the performance that was reported by Chen and Qian (2020).

Chen and Qian (2020) proposed a unified model for AE, OE and SC. They explicitly modelled relationships between the subtasks with an attention mechanism, based on BERT. Their method is the SOTA on ABSA to the best of our knowledge.

Dankers et al. (2019) proposed a hard-parameter sharing MTL model for modelling SMI and emotions from the dimension of valence, arousal and dominance, where the BERT-based valence and metaphor identification MTL model yielded the best performance. We use this method as a baseline for the SMI comparison. Dankers et al. (2019) also reported a single task learning performance based on BERT (STL-BERT).

Le, Thai, and Nguyen (2020) proposed a Graph Convolutional Neural Network-based MTL model, learning SMI and Word Sense Disambiguation. Their model is based on the concatenate features of GloVe, ELMo (Peters et al. 2018a) and index embeddings (GEI).

Model	SMI-F1	Acc.
STL-BERT$^\sharp$	76.3	-
Dankers et al. (2019)-BERT	76.8	-
Le, Thai, and Nguyen (2020)-GEI	75.1	93.8
GBM-MTL-BERT-ours	**77.6***	**94.5***

Table 4: Model performance on sequential metaphor identification. $^\sharp$ was reported by Dankers et al. (2019)

Setups

4 Transformer layers ($n = 3$) with 16 heads and 1024 dimensions are employed upon BERT-large as a task-specific tower. We do not change the shape of the output of GBM, thus, the learnt parameters W and V in GBM are also 1024 dimensions. Learning parameters in the model are initialised with PyTorch (Paszke et al. 2017) default setups. We use a batch size of 4, Adam optimiser (Kingma and Ba 2014) with an initial learning rate of 1e-5 and a Step Decay schedule. For the loss weights in E.q. 6, ω^{AE}, ω^{OE} and ω^{SC} are 0.5, 0.5 and 2, respectively[2]; ω^{SMI} and ω^{POS} are 1 and 1, respectively. The model is fine-tuned with 20 epochs on training sets. A test set result is given by a trained model that has the best performance on the validation set in terms of the main measure (F1) of each task. Because our tasks are formalised as sequence tagging tasks, where the lengths of input tokens should be equal to the lengths of labels, we take the prediction of the first WordPiece (Wu et al. 2016) token as the prediction of the original word. For ABSA, we use ABSA-F1 score (Macro-F1) from Chen and Qian (2020) as the main measure, where the sentiment prediction for an aspect term is correct in ABSA-F1, only if the predictions of AE and SC of the same term are correct. For SMI, F1 is the main measure, where metaphors are positive labels.

Results

In Table 3, our model achieves the best performance according to the main measure of ABSA-F1 on both datasets, yielding an average gain of 1.26% against the strongest baseline (Chen and Qian 2020). For AE and SC, our model also outperforms the strongest baselines of the subtasks, yielding average gains of 0.70% and 1.29%, respectively. However, our model does not outperform the strongest baselines on

[2]Since learning SC is harder than learning other subtasks, a higher loss weight is assigned to SC.

Model	Lap.	Res.	VUA	Avg.
TB_MTL	60.21	66.37	75.73	67.44
TB_MTL_GBM	61.40	67.07	76.69	68.39
TB_MTL_WSP	60.88	66.79	76.22	67.96
TB_MTL_WSP_GBM	**61.54**	**67.17**	**76.87**	**68.53**

Table 5: Ablation analysis, measured by ABSA-F1/F1 on validation sets. TB_MTL is a hard-parameter sharing model with four Transformer layers upon BERT; GBM is Gated Bridging Mechanism; WSP is weighted sum pooling.

No. of blocks	Lap.	Res.	VUA	Avg.
1	60.17	66.28	75.52	67.32
2	60.78	67.01	76.60	68.13
3	61.43	67.10	76.85	68.46
4	**61.54**	**67.17**	**76.87**	**68.53**

Table 6: Model performance with different numbers of blocks on validation sets, measured by ABSA-F1/F1.

OE. This is presumably because Chen and Qian (2020) as well as He et al. (2019) introduced different methods to ensure that a target token cannot have both aspect and opinion labels in ABSA, which improves their accuracy in the OE tasks. Chen and Qian (2020), e.g., employed a different regularisation hinge loss for AE and OE. He et al. (2019), on the other hand, considered AE-OE as a unified subtask with a different annotation paradigm.

In Table 4, our model outperforms the SMI baselines by at least 0.8% F1 score. There is a gain of 1.3% compared with the BERT-based single task learning model (STL-BERT).

Ablation Analysis

To investigate the utilities of different components in our model, we conduct an ablation analysis on the validation sets of ABSA and SMI with the following setups: (1) TB_MTL (a hard-parameter sharing MTL model with a BERT sharing encoder and four Transformer layers in each task-specific tower); (2) TB_MTL_GBM (TB_MTL with Gated Bridging Mechanism); (3) TB_MTL_WSP (TB_MTL with weighted sum pooling); (4) TB_MTL_WSP_GBM (our full model containing all of above components). As seen in Table 5, compared with TB_MTL, the application of GBM improves model performance by 0.95% on average. Besides, weighted sum pooling (TB_MTL_WSP) also provides additional gains (0.52%) on a hard-parameter sharing model. Finally, the full model performs the best, yielding a gain of 1.09% against the hard-parameter sharing model.

Next, we examine an average pooling strategy (Chen and Qian 2020) based on TB_MTL instead of using WSP. There are drops of 0.20% and 0.16% in SMI and PoS tagging F1, respectively. We also observe drops of 0.29% and 0.18% in ABSA-F1 in laptop and restaurant datasets, respectively. These drops show that the weighted sum pooling strategy can dynamically learn from strong layers, hence yielding better performance compared with the average pooling that has fixed weights for different pooling layers.

Finally, we test the impact of different numbers of blocks in our full model (TB_MTL_WSP_GBM) based on the validation sets of ABSA and SMI. For a model that only has a bloke (Block 0), we do not employ WSP or GBM. A task-specific tower only consists of a Transformer layer and a linear layer. As seen in Table 6, the performance of the model can be improved by increasing the number of blocks; however, the average improvement of the model with four blocks is small (0.07%) compared to the model with three blocks. Generally, it can be seen that GBM and WSP can provide better learning capabilities to the MTL model as a model becomes deeper.

Comparison between Different Bridges

Previously, several information transformation mechanisms were proposed for MTL on different tasks, e.g., cross-stitch (Misra et al. 2016), attention (Chen and Qian 2020), and vanilla gating mechanisms (Dankers et al. 2019). We compare our GBM against these methods based on the same backbone of BERT and Transformers. Hard-parameter sharing is introduced as a baseline. We examine these methods with the validation sets of laptop, restaurant and VUA.

Cross-stitch (Misra et al. 2016) is a linear information transformation method. The fused features in a task-specific tower are given by the weighted sum of information from the private tower and neighbour towers, where the weights are learnt parameters. Since the weight of a hidden state vector is a real number, the implicit hypothesis of cross-stitch is that all elements of a hidden state from a neighbour tower are equally significant for fusing with the hidden state in a private tower. Attention is another widely applied information transformation method (Liu, Johns, and Davison 2019; Liu et al. 2019b; He et al. 2019; Chen and Qian 2020), where the attention weights of a neighbour tower are given by a linear scoring function and Softmax normalisation. The attended hidden states are given by the weighted sum of hidden states from a neighbour tower. The implicit hypothesis of an attention-based method is that at least one hidden state vector from a neighbour tower is supportive for a private tower, because the attention weights are normalised based on the hidden states in the neighbour tower. Since there are different modifications of attention for the fitting of different tasks, we employ the method of Chen and Qian (2020) for benchmark. The fused features are given by the concatenation of the hidden states from a private tower and the attended information from neighbour towers. We employ a sigmoid-activated fully-connected layer to project the concatenated features to 1024 dimensions as the soft-parameter sharing output. A vanilla gating mechanism from Dankers et al. (2019) simply uses an update gate for fusing information from private and neighbour towers, where the update gate is controlled by the concatenation of hidden states from a private tower and a neighbour tower. This method does not have a reset gate and non-linear projection on the neighbour tower hidden states. Thus, the implicit hypothesis is that hidden states from a neighbour tower and hidden states from a private tower are in the same vector space. We employ E.q. 11 to incorporate the fusion of multiple subtasks.

In contrast, we employ gating mechanisms and a non-

Bridge	Lap.	Res.	VUA	Avg.
Hard-param. sharing	60.21	66.37	75.73	67.44
+ Cross-stitch	60.19	66.46	75.78	67.48
+ Attention	60.61	66.84	76.03	67.83
+ Vanilla Gating	60.38	66.52	76.27	67.72
+ GBM-ours	**61.40**	**67.07**	**76.69**	**68.39**

Table 7: Different information transformation mechanism performance on validation sets, measured by ABSA-F1/F1.

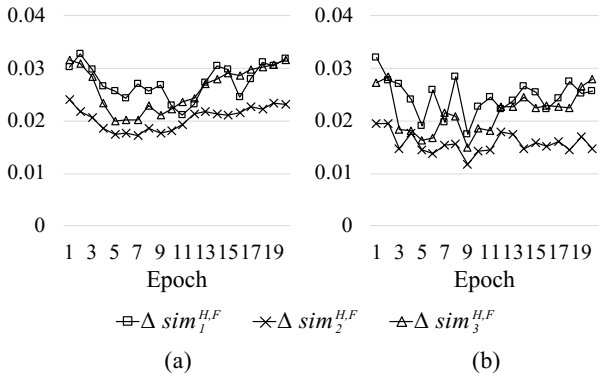

Figure 3: Gated Bridging Mechanism utility analysis. $\Delta sim_i^{H,F} = sim_{i,fake}^{H,F} - sim_{i,real}^{H,F}$, where $sim_{i,fake}^{H,F}$ and $sim_{i,real}^{H,F}$ are cosine similarities between pre-fused (H) and post-fused (F) features in Block i in the SMI task-specific tower. The subscript $fake$ indicates that the hidden states from an auxiliary task are randomly generated; while the subscript $real$ indicates that the auxiliary hidden states are from a real PoS tagging tower. (a) is based on VUA training set; (b) is based on the validation set.

linear projection on the filtered hidden states from a neighbour tower (see E.q. 8). The hypotheses of GBM are that (1) hidden states that are from a neighbour tower are possibly useless for specific private tower learning. The use of gating mechanisms should filter out the useless hidden states. (2) Hidden states in different towers are not in the same vector space. The use of a non-linear projection layer can project the vector space of a neighbour tower to the vector space of a private tower.

As seen in Table 7, GBM outperforms other information transformation mechanisms by at least 0.56% on average. The use of reset gate and non-linear projection on the neighbour tower hidden states yields an average gain of 0.67% compared with vanilla gating. Generally, attention is more effective than other baseline methods, although vanilla gating shows a moderately improved performance on the VUA dataset. Finally, all of these methods surpass the hard-parameter sharing baseline.

Gated Bridging Mechanism Analysis

We introduce a probing task for SMI to examine if GBM can reject useless information from randomly generated hidden states and fuse useful information from the hidden states of a real PoS tagging tower. Based on a hard-parameter sharing model with two task specific towers (TB_MTL in Table 5), we employ GBM in the SMI tower. Thus, GBM can transfer information from the PoS tagging tower to the SMI tower in each block. For the baseline model, we use randomly generated hidden states instead of the hidden states that are learnt from the PoS tagging task tower to indicate that the auxiliary task does not yield supportive hidden states for the SMI task. We hypothesise that GBM can reject the random hidden states during the SMI learning.

The hypothesis is tested by comparing cosine similarity between pre-fused Transformer hidden states (H_{i-1}^{SMI}) and post-fused features (F_i^{SMI}, see E.q. 10) in the tower of SMI, where $i \in \{1, 2, 3\}$. The cosine similarity for matrices is given by $sim_i^{H,F} = cosine(H_{i-1}^{\text{SMI}}, F_i^{\text{SMI}})$, averaged over sequences in the VUA training and validation sets. A lower $sim_i^{H,F}$ indicates that F_i^{SMI} has gained more information from the PoS tagging tower, because F_i^{SMI} is more distinct from H_{i-1}^{SMI} after H_{i-1}^{SMI} and H_{i-1}^{PoS} are passed through GBM in the SMI tower. On the other hand, a higher $sim_i^{H,F}$ signifies that GBM of the SMI tower has rejected more information from the PoS tagging tower (H_{i-1}^{PoS}) to fuse with H_{i-1}^{SMI}, because the post-fused F_i^{SMI} and pre-fused H_{i-1}^{SMI} are more similar. We use $\Delta sim_i^{H,F}$ to indicate the fused information gap of SMI by learning from the PoS tagging auxiliary task and the random hidden states, where $\Delta sim_i^{H,F} = sim_{i,fake}^{H,F} - sim_{i,real}^{H,F}$.

As seen in Figure 3, $\Delta sim_i^{H,F}$ remains in positive intervals on both VUA training and validation sets. It shows that GBM has fused more information from the PoS tagging hidden states and rejecting more information from the randomly generated hidden states across all blocks. Besides, according to Liu et al. (2019a), high-level DNN layers encode task-specific features, while low-level layers encode general features. The observation that $\Delta sim_1^{H,F}$ and $\Delta sim_3^{H,F}$ are on average larger than $\Delta sim_2^{H,F}$ shows that a supportive auxiliary task provides more useful general and task-specific features for the main task in MTL.

Conclusion

We propose a novel Gated Bridging Mechanism (GBM) for soft-parameter sharing between multi-task learning (MTL) towers. Based on a Transformer backbone and GBM, our MTL model outperforms previous baselines on aspect-based sentiment analysis and sequential metaphor identification tasks. GBM also yields better performance than previous soft-parameter sharing methods, e.g., cross-stitch, attention and vanilla gating based on the same backbone. We provide insight into hidden states of MTL task-specific towers, showing that the proposed GBM can functionally gain useful knowledge from the hidden states of a supportive auxiliary task, and rejecting useless hidden states. Finally, applying the weighted sum pooling strategy further improves our model performance, because it dynamically learns features from effective layers in an MTL model for each subtask.

References

Chen, X.; Leong, C. W.; Flor, M.; and Klebanov, B. B. 2020. Go figure! Multi-task Transformer-based architecture for metaphor detection using idioms: ETS team in 2020 metaphor shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, 235–243.

Chen, Z.; and Qian, T. 2019. Transfer capsule network for aspect level sentiment classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 547–556.

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

Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734.

Cipolla, R.; Gal, Y.; and Kendall, A. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7482–7491. IEEE.

Dankers, V.; Rei, M.; Lewis, M.; and Shutova, E. 2019. Modelling the interplay of metaphor and emotion through multitask learning. 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)*, 2218–2229.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.

Dong, D.; Wu, H.; He, W.; Yu, D.; and Wang, H. 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 (Volume 1: Long Papers)*, 1723–1732.

Duong, L.; Cohn, T.; Bird, S.; and Cook, P. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 845–850.

Gao, G.; Choi, E.; Choi, Y.; and Zettlemoyer, L. 2018. Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Gong, H.; Gupta, K.; Jain, A.; and Bhat, S. 2020. Illinimet: Illinois system for metaphor detection with contextual and linguistic information. In *Proceedings of the Second Workshop on Figurative Language Processing*, 146–153.

Graves, A.; and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18(5-6): 602–610.

Guo, P.; Lee, C.-Y.; and Ulbricht, D. 2020. Learning to branch for multi-task learning. In *Proceedings of the 37th International Conference on Machine Learning*.

He, R.; Lee, W. S.; Ng, H. T.; and Dahlmeier, D. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 504–515.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Honnibal, M.; and Montani, I. 2017. spaCy 2: Natural language understanding with Bloom Embeddings. *Convolutional Neural Networks and Incremental Parsing* .

Hu, M.; Peng, Y.; Huang, Z.; Li, D.; and Lv, Y. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 537–546.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Le, D.; Thai, M.; and Nguyen, T. 2020. Multi-task learning for metaphor detection with graph convolutional neural networks and word sense disambiguation. In *AAAI*, 8139–8146.

Li, X.; Bing, L.; Lam, W.; and Shi, B. 2018. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 946–956.

Li, X.; Bing, L.; Li, P.; and Lam, W. 2019a. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6714–6721.

Li, Z.; Li, X.; Wei, Y.; Bing, L.; Zhang, Y.; and Yang, Q. 2019b. Transferable End-to-End Aspect-based Sentiment Analysis with Selective Adversarial Learning. 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)*, 4582–4592.

Liu, N. F.; Gardner, M.; Belinkov, Y.; Peters, M. E.; and Smith, N. A. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1073–1094.

Liu, P.; Fu, J.; Dong, Y.; Qiu, X.; and Cheung, J. C. K. 2019b. Learning multi-task communication with message passing for sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4360–4367.

Liu, P.; Qiu, X.; and Huang, X. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2873–2879.

Liu, S.; Johns, E.; and Davison, A. J. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1871–1880.

Long, M.; Cao, Z.; Wang, J.; and Yu, P. S. 2017. Learning multiple tasks with multilinear relationship networks. In *Advances in Neural Information Processing Systems 30*, 1593–1602.

Luo, H.; Li, T.; Liu, B.; and Zhang, J. 2019. DOER: Dual cross-shared RNN for aspect term-polarity co-extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 591–601.

Mao, R.; Lin, C.; and Guerin, F. 2018. Word Embedding and WordNet Based Metaphor Identification and Interpretation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1222–1231.

Mao, R.; Lin, C.; and Guerin, F. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3888–3898.

Marcus, M. P.; Santorini, B.; and Marcinkiewicz, M. A. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2): 313–330.

Misra, I.; Shrivastava, A.; Gupta, A.; and Hebert, M. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3994–4003.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language processing*, 1532–1543.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018a. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2227–2237.

Peters, M.; Neumann, M.; Zettlemoyer, L.; and Yih, W.-t. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1499–1509.

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. SemEval-2014 Task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 27–35. Dublin, Ireland: Association for Computational Linguistics.

Ruder, S. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* .

Ruder, S.; Bingel, J.; Augenstein, I.; and Søgaard, A. 2017. Sluice networks: Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142* .

Ruder, S.; Bingel, J.; Augenstein, I.; and Søgaard, A. 2019. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4822–4829.

Shutova, E.; Kiela, D.; and Maillard, J. 2016. Black Holes and White Rabbits: Metaphor Identification with Visual Features. *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 160–170.

Steen, G. J.; Dorst, A. G.; Herrmann, J. B.; Kaal, A.; Krennmayr, T.; and Pasma, T. 2010. *A Method for Linguistic Metaphor Identification: From MIP to MIPVU*, volume 14. John Benjamins Publishing.

Su, C.; Fukumoto, F.; Huang, X.; Li, J.; Wang, R.; and Chen, Z. 2020. DeepMet: A reading comprehension paradigm for token-level metaphor detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, 30–39.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, W.; Pan, S. J.; Dahlmeier, D.; and Xiao, X. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Wu, C.; Wu, F.; Chen, Y.; Wu, S.; Yuan, Z.; and Huang, Y. 2018. Neural metaphor detecting with CNN-LSTM model. In *Proceedings of the Workshop on Figurative Language Processing*, 110–114.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Xu, H.; Liu, B.; Shu, L.; and Philip, S. Y. 2018. Double embeddings and CNN-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 592–598.

Zhang, Y.; and Yang, Q. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* .

Zheng, Y.; Zhang, R.; Wang, S.; Mensah, S.; and Mao, Y. 2020. Anchored Model Transfer and Soft Instance Transfer for Cross-Task Cross-Domain Learning: A Study Through Aspect-Level Sentiment Classification. In *Proceedings of The Web Conference 2020*, 2754–2760.