Anchored Model Transfer and Soft Instance Transfer for Cross-Task Cross-Domain Learning: A Study Through Aspect-Level Sentiment Classification

Yaowei Zheng
BDBC and SKLSDE, School of Computer Science and Engineering, Beihang University
Beijing, China

Richong Zhang*
BDBC and SKLSDE, School of Computer Science and Engineering, Beihang University
Beijing, China

Suyuchen Wang
BDBC and SKLSDE, School of Computer Science and Engineering, Beihang University
Beijing, China

Samuel Mensah
BDBC and SKLSDE, School of Computer Science and Engineering, Beihang University
Beijing, China

Yongyi Mao
School of Electrical Engineering and Computer Science, University of Ottawa
Ottawa, Canada

ABSTRACT

Supervised learning relies heavily on readily available labelled data to infer an effective classification function. However, proposed methods under the supervised learning paradigm are faced with the scarcity of labelled data within domains, and are not generalized enough to adapt to other tasks. Transfer learning has proved to be a worthy choice to address these issues, by allowing knowledge to be shared across domains and tasks. In this paper, we propose two transfer learning methods Anchored Model Transfer (AMT) and Soft Instance Transfer (SIT), which are both based on multi-task learning, and account for model transfer and instance transfer, and can be combined into a common framework. We demonstrate the effectiveness of AMT and SIT for aspect-level sentiment classification showing the competitive performance against baseline models on benchmark datasets. Interestingly, we show that the integration of both methods AMT+SIT achieves state-of-the-art performance on the same task.

CCS CONCEPTS

• **Computing methodologies** → **Transfer learning**; • **Information systems** → *Sentiment analysis.*

KEYWORDS

transfer learning, sentiment analysis

ACM Reference Format:
Yaowei Zheng, Richong Zhang, Suyuchen Wang, Samuel Mensah, and Yongyi Mao. 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 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380034

*Corresponding author: zhangrc@act.buaa.edu.cn

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380034

1 INTRODUCTION

Deep learning has demonstrated great successes in some areas of supervised learning, when a large amount of labelled data (such as ImageNet) were made available. However, in many applications, obtaining labelled training data can be quite difficult or expensive, and one must work with a small set of labelled data. This significantly limits the predictive performance that can be achieved by a machine learning model. Among other approaches, transfer learning [40] is an important methodology that allows one to go beyond this limitation. In transfer learning, this is achieved by utilizing models or data in another domain or task and transferring the knowledge therefrom. Transfer learning has been used successfully to a wide spectrum of applications, including, e.g., machine translation [15, 38], speech recognition [53], and aspect-level sentiment classification [9, 22, 31], etc.

In general, the objective of transfer learning can be formulated as transferring knowledge from one task to another task, with the former referred to as the *source task* and the latter as the *target task*. There is a rich body of literature the techniques for transfer learning based on deep neural networks. These techniques can be categorized into the following categories: model transfer, domain adaptation, and instance transfer. Model transfer often arises in multi-task learning, where predictive models need to be learned from several related tasks simultaneously. In that context, the model transfer approach makes certain model components shared across the different tasks [3]. In instance transfer, training examples in one task are also used in part for the training of model in the other task [2]. In domain adaptation, a common representation is learned for input examples from different tasks [17, 20, 43, 61].

We note that in model transfer, the inputs to a shared model component are assumed to have the same statistics, although they may come from different tasks. In particular, when the shared component directly takes a training instance as its input, the underlying assumption is that the data of the source task is from the same domain as the data from the target task. Instance transfer and domain

adaptation both deal with the setting where the source domain and target domain are different. In domain adaptation, the underlying assumption is that the tasks of interest are *similar*, for example, in the sense that a common classifier can be applied to the common representation of instances from both tasks. In instance transfer, the assumption is that the source and target domains have some similarity and the selection of instances for transfer is decided heuristically based on certain prior knowledge on this similarity.

In practice, transfer learning is often complicated by scenarios where none of these approaches adequately addresses the nature of the learning problem. Specifically, a considerable discrepancy may exist between the source task and the target task and between the source domain and the target domain. For example, when taking aspect-level sentiment classification as the target task, we may take a document-level sentiment classification task as the source task, while the two tasks may involve data from two different domains.

Motivated by this observation, the thrust of this paper is to develop transfer learning techniques that automatically adapt to task differences and domain differences. To that end, two techniques, referred to as Anchored Model Transfer (AMT) and Soft Instance Transfer (SIT), are developed in this paper. Specifically, the two methods are designed to automatically adapt to the task difference and domain difference. In AMT, the source and target models each have a component that "anchors" on a common model but also "drifts" away from the model. The drifts of the source and target models from the "anchor" as well as the distinct components of the models then account for the difference of the two tasks. On the other hand, In SIT, instances from the source task are selected automatically and used for the training of the target model. It is worth noting, that unlike the previous instance transfer methods, the selection of instances in SIT is probabilistic (or "soft") during the training process.

It is also worth pointing out that the proposed AMT and SIT techniques can be combined in a common framework to simultaneously deal with both issues. Taking aspect-level sentiment classification (ASC) as the target task, and document-level sentiment classification (DSC) as the source task, we perform extensive experiments to demonstrate the application of these techniques. The effectiveness of these techniques is shown on common benchmark datasets, and new state-of-the-art results are obtained.

This paper is organized as follows. In Section 2, we present related works for the study under transfer learning and ASC. In Section 3, we present our proposed methods AMT and SIT for transfer learning. In Section 4, we perform extensive experiments showing the effectiveness of our proposed method. We give concluding remarks in Section 5.

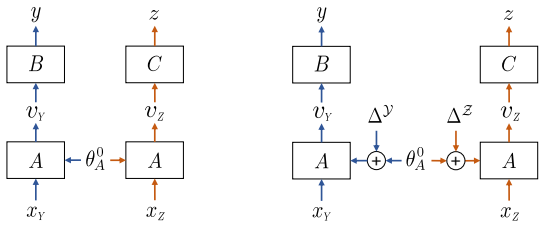
2 RELATED WORK

In recent years, transfer learning has become a popular approach to improve learning in a target task. Pertaining to the knowledge being transferred, transfer learning approaches can be classified into mainly three categories; (1) model transfer [4, 6, 14, 24, 39, 57, 59], (2) instance transfer [11, 29, 32, 60] and (3) domain adaptation [17, 20, 61, 61]. These strategies allow knowledge to be shared across domains or tasks. Recent approaches have also explored domain adaptation through adversarial networks [1, 17, 54].

Multi-Task Learning (MTL) learns multiple tasks simultaneously by sharing model parameters to share knowledge across tasks and domains. We find MTL across several NLP domains including sequence labeling [14, 57, 59], implicit discourse relation classification [33], and machine translation [38] among others. [4] allows each task to have its own model parameters, but the parameters are regularized to encourage generalization on the target task for flexibility and proposed a dynamic transfer network which exploits a gated architecture to find a suitable set of parameter configurations.
Domain Adaptation attempts to bridge the gap between the marginal distributions of the two domains by learning domain-invariant structures. Deep neural networks [7, 17, 20, 34, 58, 61, 62] have been explored for domain adaptation due to its ability to learn expressive representations that are transferable between similar tasks. Some recent approaches have also employed a domain-adversarial loss for domain adaptation [1, 17, 54]. To a considerable degree, works under domain adaptation have considered only to learn a model for the same task across domains. The MTL approach to domain adaptation can address multiple tasks [18, 20, 42].
Instance Transfer leverages the sufficient instances of the source domain to support the insufficient data in the target domain. Several boosting-based algorithms [11, 16, 29, 32, 60] have been proposed for instance transfer. Some notable efforts have also been made for instance transfer in regression problems [41]. Unlike transferring instance of labelled data, self-training [37] on the other hand takes advantage of the model's predictions on unlabelled data to support learning of the target task. Based on the assumption that MTL attends to the domain shift problem, self-training is more simple and in line for instance transfer in this paper. We find its applications in areas such as parsing [45] and sentiment analysis [23] and intend to apply it for instance transfer to improve the objective predictive function.
Aspect-Level Sentiment Classification aims at identifying the sentiment polarity (e.g. positive, negative, neutral) of the sentence concerning the given aspect. Generally, we have a set of aspect-sentence pairs, where the aspect is a specific span occurring in the sentence. Contemporary approaches for ASC are based on neural networks [8, 9, 12, 13, 21, 22, 28, 30, 31, 35, 48, 51, 52, 55, 56], shelving traditional approaches [25, 27] which heavily depends on human-engineered features and leads to error propagation and performance degradation. The recent interest in neural networks is attributed to its ability to learn rich representations that capture the interactions between an aspect term and contextual words. For the insufficient amount of labelled data in ASC domain, transfer learning has been investigated [9, 22, 31], allowing knowledge to be transferred from the DSC domain which has a vast amount of labelled data to the ASC domain with insufficient data. These methods suggest how related the ASC domain and the DSC domain and serve as an opening for further research in transfer learning for ASC.

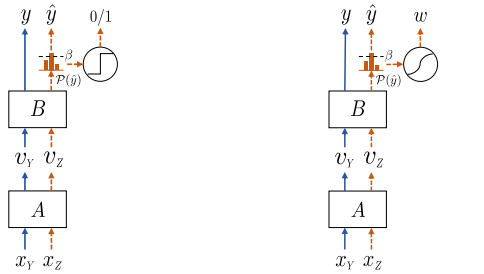
3 METHODS

In this paper, we study the art of solving the target task by transferring knowledge from another auxiliary task. Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be the input space, the output space of the target task and the output space of the auxiliary task respectively. For simplicity, we call the task of



(a) Vanilla Model Transfer. (b) Anchored Model Transfer.

Figure 1: Two methods for model transfer.



(a) Hard Instance Transfer. (b) Soft Instance Transfer.

Figure 2: Two methods for instance transfer.

predicting Y and Z for some input X "$\mathcal{Y}$-task" and "$\mathcal{Z}$-task" respectively. The two different tasks may have different output. But to transfer knowledge from the auxiliary task to support the learning in the target task, it is assumed that the two tasks are related. In our setting, they have the same input space, although the distributions on the common input space for different tasks may be different. For example, to accomplish the aspect-level sentimental classification task, we may exploit the vast amount of document-level labelled data. The $\mathcal{Y}$-task is ASC. The $\mathcal{Z}$-task is DSC. And the texts make up the same input space $\mathcal{X}$.

Note that to fully take advantage of transfer learning, the input of the auxiliary task may come from a different domain from the target task. For example, the knowledge from the sentiment classification of laptop reviews may be helpful for the sentiment classification of restaurant reviews. Let $\mathcal{X}_Y$ and $\mathcal{X}_Z$ be the set of input instances in $\mathcal{Y}$-task and $\mathcal{Z}$-task respectively. Formally, two domains refer to two different input distributions $\mathcal{P}(X)$ on the set $\mathcal{X}_Y$ and $\mathcal{X}_Z$.

In the following subsections, we will introduce two ways of transferring: Anchored Model Transfer and Soft Instance Transfer for cross-domain cross-task learning. The aspect-level sentiment classification task is studied as an example.

3.1 Anchored Model Transfer

The model transferring approach solves the target task and the auxiliary task simultaneously, while the models for two tasks share some modules, denoted as A. Assume that the sub-model A maps the same input space $\mathcal{X}$ to an internal representation space $\mathcal{V}$, namely $A : \mathcal{X} \to \mathcal{V}$. Besides the common module A, let the left upper modules of the two models be B and C for the two tasks respectively. Module B and C takes the internal representation V as input, and finally output the label distributions $\mathcal{P}(Y), \mathcal{P}(Z)$ over output space.

Model T :

$$A : \mathcal{X} \to \mathcal{V}, \qquad B : \mathcal{V} \to \mathcal{P}(\mathcal{Y})$$

Model S :

$$A : \mathcal{X} \to \mathcal{V}, \qquad C : \mathcal{V} \to \mathcal{P}(\mathcal{Z})$$

Although the two tasks share the same module A, their parameters, denoted as $\theta_A^{\mathcal{Y}}$ and $\theta_A^{\mathcal{Z}}$ correspondingly, are not necessarily the same. We will in fact consider that there is a θ_A^0 such that

$$\begin{aligned} \theta_A^{\mathcal{Y}} &= \theta_A^0 + \Delta^{\mathcal{Y}} \\ \theta_A^{\mathcal{Z}} &= \theta_A^0 + \Delta^{\mathcal{Z}} \end{aligned}$$

where $\Delta^{\mathcal{Y}}$ and $\Delta^{\mathcal{Z}}$ are small disturbance variables following a normal distribution with a zero mean. This formulation implies that we assume the marginal distributions on the feature space of the input in both tasks to be approximate. Here the common model parameter θ_A^0 serves as an anchor. The two models are "anchored" on the same base model with parameter θ_A^0. We refer this knowledge sharing over the model as Anchored Model Transfer which is illustrated in Figure 1.

Under this formalization, for any input $x \in \mathcal{X}_Y$ of the target task, the output of module A and B can be denoted as $A(x; \theta_A^0 + \Delta^{\mathcal{Y}})$ and $\mathcal{P}(\hat{Y}) = B(A(x; \theta_A^0 + \Delta^{\mathcal{Y}}); \theta_B)$ respectively, where θ_B is the parameter in module B. Similarly, the output of the auxiliary task can be denoted as $\mathcal{P}(\hat{Z}) = C(A(x; \theta_A^0 + \Delta^{\mathcal{Z}}); \theta_C)$ where $x \in \mathcal{X}_Z$ is the input and θ_C is the parameter in module C. The model transferring approach simultaneously learns the two tasks by optimizing the joint losses of the two tasks over all model parameters $(\theta_A^0, \Delta^{\mathcal{Y}}, \Delta^{\mathcal{Z}}, \theta_B, \theta_C)$. The loss over all training samples for two classification tasks is $\ell_{\mathcal{Y}+\mathcal{Z}}$.

$$\ell_{\mathcal{Y}+\mathcal{Z}} := \sum_{(x,y)\in\mathcal{D}_{XY}} \mathrm{CE}\left(\mathcal{P}(\hat{Y}), y\right) + \sum_{(x,z)\in\mathcal{D}_{XZ}} \mathrm{CE}\left(\mathcal{P}(\hat{Z}), z\right) + \lambda_1 \|\Delta^{\mathcal{Y}}\|^2 + \lambda_2 \|\Delta^{\mathcal{Z}}\|^2 \quad (1)$$

where $\mathrm{CE}(\cdot)$ denotes the cross-entropy loss, which is equivalent to the negative log-likelihood. λ_1 and λ_2 are the L_2 regularization constants. Since λ controls to what extent the shared part of the model can "drift" from the anchored model parameters, it can be called the model drift coefficient.

3.2 Soft Instance Transfer

Instance transfer approach makes use of the instances in the auxiliary task for the training of the target task. The basic idea here is similar to "pseudo-labelling" used in semi-supervised learning: If a training example $x \in \mathcal{X}_Z$ in the auxiliary $\mathcal{Z}$-task is also useful for the main $\mathcal{Y}$-task, it may serve also as an example for the $\mathcal{Y}$-task although we have never observed its label in the $\mathcal{Y}$-task. The hypothesis here is that there are a significant number of examples in the $\mathcal{Z}$-task having such a nature. Under this hypothesis, we can actually use the $\mathcal{Y}$-task model to predict the label for x in the $\mathcal{Z}$-task; if the prediction is quite confident, we regard the prediction as correct, and use the prediction as the true label for x in the $\mathcal{Y}$-task.

Table 1: The statistics for datasets.

Task	Dataset		Positive	Negative	Neutral
ASC	Rest14	train	2164	807	637
		test	728	196	196
	Laptop	train	994	870	464
		test	341	128	169
	Rest16	train	1620	190	88
		test	597	709	38
DSC	Yelp	train	10k	10k	10k
	Elec	train	10k	10k	10k

Formally, consider taking $x \in \mathcal{X}_Z$ as the input of model T for $\mathcal{Y}$-task. The output $\mathcal{P}(\hat{Y}) = T(x; \theta_T)$ is the distribution over $\mathcal{Y}$. Maximizing this output probability over all label values, we get the predicted label $\hat{y} = \arg\max \mathcal{P}(\hat{Y})$ with probability $\mathcal{P}(\hat{y}) = \max \mathcal{P}(\hat{Y})$. We define the loss weight assigned to each auxiliary instance as follows:

$$w(x) = \sigma\left(\alpha \cdot (\mathcal{P}(\hat{y}) - \beta)\right)$$

where β is a hyperparameter indicating a probability threshold, σ is the *sigmoid* function, and $\sigma(\alpha(\cdot))$ is a smooth approximation of the step function. α controls the smoothness of w's curve, so we call it the smoothness coefficient. The loss weight w is designed to let the model tend to focus on those auxiliary instances that have higher predict confidence, while for instances with confidence slightly lower than the threshold, the model can still make use of them by assigning a relatively lower weight. With this weight $w(x)$, we can partly use the instance of $\mathcal{Z}$-task to train the model for $\mathcal{Y}$-task. We therefore refer this approach with instance weighting as Soft Instance Transfer as shown in Figure 2. Based on the definition of instance weight, we define the following additional loss terms:

$$\ell_{\mathcal{Z}\to\mathcal{Y}} := \sum_{x\in\mathcal{X}_Z} w(x)\mathrm{CE}\left(\mathcal{P}(\hat{Y}), \hat{y}\right)$$

The soft instance transfer learning approach optimizes $\ell_{\mathcal{Y}} + \eta\ell_{\mathcal{Z}\to\mathcal{Y}}$ over parameters θ_T to achieving a high performance in the $\mathcal{Y}$ task, where η controls the influence of the additional loss term.

Note that the Anchored Model Transfer and Soft Instance Transfer can be integrated to jointly transfer knowledge from one task to the other by optimizing:

$$\ell_{\mathcal{Y}+\mathcal{Z}} + \eta_1\ell_{\mathcal{Z}\to\mathcal{Y}} + \eta_2\ell_{\mathcal{Y}\to\mathcal{Z}} \tag{2}$$

3.3 The Baseline Model

In this paper, we demonstrate the application of the proposed two ways of transfer learning and their integration to the ASC task. A DSC task or ASC task is served as an auxiliary task. We describe the models for the ASC task and auxiliary tasks in detail. The common module A shared by two tasks consists of a pre-trained word embedding layer, a Bi-GRU layer and a mean-pooling layer. For the separated module B and C of the target task and auxiliary task, we just adopt a softmax layer in each module. The input of both ASC and DSC tasks are sentences. Each input sentence can be denoted as $x = \{x_1, x_2, \cdots, x_n\}$, where n is the sentence length. For ASC tasks, each sentence may contain multiple aspects. In order to mark the aspect in the sequence, an aspect location label is added in the sequence right in front of and after the aspect term. The shared embedding layer maps each sentence x to a word embedding sequence $\boldsymbol{E}_x = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n\}$ through a pre-trained lookup table $\boldsymbol{E} \in \mathbb{R}^{d_w \times |V|}$ where each column of $\boldsymbol{E}$ is the embedding of a word in the vocabulary, d_w is the dimension of word embedding vectors, and $|V|$ is the vocabulary size. Then, we pass the embedded sequence E_x into a single layer bidirectional GRU (Bi-GRU) module [10] to utilize the contextual information. The hidden state of each time step is the concatenation of the hidden states in two stacked GRUs that process the sequence in the opposite direction. After that, a mean-pooling layer [49] is employed over the Bi-GRU's hidden states to generate the representation of the whole sentence. The last layer is a softmax which takes the output $\boldsymbol{v} \in \mathcal{V}$ of the common module A as input and outputs the predictive distribution $\mathcal{P}(\hat{Y})$ or $\mathcal{P}(\hat{Z})$ over the sentiment polarity towards a certain aspect (in ASC tasks) or the full sentence (in DSC tasks).

Table 2: Transfer methods' performance on all dataset pairs.

Method	Rest14		Laptop		Rest16	
	ACC	F1	ACC	F1	ACC	F1
Base	80.54	70.13	75.04	70.26	87.52	67.18
Cross Task						
AMT	81.87	73.09	76.78	72.33	88.97	70.86
SIT	81.50	72.79	76.48	71.72	88.61	70.32
AMT+SIT	**82.47**	**73.60**	**77.02**	**72.52**	**89.45**	71.35
Cross Domain						
AMT	81.59	72.81	76.18	72.08	88.62	70.50
SIT	81.25	71.65	75.60	71.03	88.36	68.65
AMT+SIT	82.14	73.20	76.80	72.71	89.09	**71.49**
Cross Task & Domain						
AMT	81.33	71.14	76.05	71.33	88.48	69.41
SIT	80.71	70.59	75.55	70.87	88.08	68.53
AMT+SIT	82.09	73.09	76.49	72.11	88.97	71.67

4 EXPERIMENTS

4.1 Datasets

Datasets for ASC We collect three datasets derived from SemEval 2014 Task 4 [47] and SemEval 2016 Task 5 [46] for the ASC task, which contain domain-specific user reviews towards restaurants (*Rest14, Rest16*) and laptop (*Laptop*) domains. Each review is accompanied with either a *positive*, *neutral* or *negative* label, expressing the sentiment towards a specific aspect in the text. Following previous works [8, 9, 22, 51], we discard instances with polarity *conflict*, where contradictory sentiments are expressed towards the same aspect. The three datasets all have pre-defined train and test splits. Note that experiments are also conducted using the three ASC datasets as auxiliary tasks with the aim to evaluate the model's performance on model transfer and instance transfer.

Datasets for DSC Datasets used for DSC tasks are gathered from Yelp Review (*Yelp*) [50] and Amazon Electronics (*Elec*) [36]. The instances of these two datasets contain user reviews towards restaurants and electronics domains respectively, and each user review

Table 3: Performance comparison of different methods on the benchmark datasets. "#": retrieved from [22]. "♮": produced with our implementation. "-": not reported. Other compared results are retrieved from the original paper.

Method	Rest14		Laptop		Rest16	
	ACC	F1	ACC	F1	ACC	F1
LSTM [48]	75.23#	64.21#	66.79#	64.02#	81.95#	58.11#
TD-LSTM [48]	75.37#	64.51#	68.25#	65.96#	82.16#	54.21#
ATAE-LSTM [56]	78.60#	67.02#	68.88#	65.93#	83.77#	61.71#
MemNet [51]	76.87#	66.40#	68.91#	62.79#	83.04#	57.91#
RAM [8]	78.48#	68.54#	72.08#	68.43#	83.88#	62.14#
MGAN [13]	81.25	71.94	75.39	72.47	84.36♮	63.20♮
TNet [30]	80.69	71.27	76.54	71.75	86.18♮	65.16♮
PRET+MULT [22]	79.11	69.73	71.15	67.46	85.58	69.76
TransCap [9]	79.55	71.41	73.87	70.10	-	-
MGAN [31]	81.66	71.72	**77.62**	72.26	-	-
Base	80.54	70.13	75.04	70.26	87.52	67.18
AMT	81.87	73.09	76.78	72.33	88.97	70.86
SIT	81.50	72.79	76.48	71.72	88.61	70.32
AMT+SIT	**82.47**	**73.60**	77.02	**72.52**	**89.45**	**71.49**

comes with a 5-level rating score. We consider a three-class classification task on the DSC datasets by artificially classifies rating score < 3, $= 3$, > 3 as *negative*, *neutral* and *positive* respectively. Note that *Yelp* shares the similar text domain with *Rest14* and *Rest16*, and *Elec* shares the similar text domain with *Laptop*.

In our experiment setup, we design different dataset pairs to evaluate the Anchored Model Transfer (AMT), Soft Instance Transfer (SIT), and a combination of both (AMT+SIT). Each pair has a target dataset and an auxiliary dataset denoted as {target, auxiliary}. The dataset pairs are divided into three classes:

- **Cross Task**: This class contains the combinations of {*Rest14, Yelp*}, {*Laptop, Elec*} and {*Rest16, Yelp*}, which have the instances with similar domains. Aims to verify the model's transferability across different tasks with similar domains;
- **Cross Domain**: This class contains the combinations of {*Rest14, Laptop*}, {*Laptop, Rest14*} and {*Rest16, Laptop*}, which are dataset pairs with a domain shift but under the same ASC task. Aims to verify the model's transferability across different domains with the same tasks;
- **Cross Task & Domain**: This class contains the combinations of {*Rest14, Elec*}, {*Laptop, Yelp*} and {*Rest16, Elec*}. Aims to verify the model's transferability when the tasks and domains are both different.

We only evaluate the model on target dataset's test set. The statistics of all the datasets mentioned above are in Table 1.

4.2 Implementation Details

In our experiments, we use the 300-dimension GloVe vectors [44] to initialize the word embeddings. The word embeddings are not updated during the training process. Parameters of all layers excluding embeddings are randomly initialized with xavier uniform initializer [19]. The dimension of the hidden states of Bi-GRU is 300. Adam optimizer [26] is used at a learning rate of 0.001. To alleviate overfitting, an L_2 regularization is added and dropout is applied with dropout rate 0.5. For simpleness, we unify λ_1 and λ_2 in Eq.(1) into a single hyperparameter λ, and unify η_1 and η_2 in Eq.(2) into a single hyperparameter η. Hyperparameter β is a scalar between 0 and 1. η is linearly increased to a preset value from zero with each epoch. The hyperparameters are tuned for each dataset pair separately. The model is trained for 100 epochs without early stopping. The batch size of both target task and auxiliary task is 64.

4.3 Main Results

To demonstrate the effectiveness of our proposed methods, we evaluate the single baseline model (Base), and the baseline model with AMT, SIT, and AMT+SIT on three benchmark datasets. The main results of the experiments are presented in Table 2. Due to the imbalance of label distribution on the datasets used, both accuracy and macro-F1 are used as evaluation metrics.

We can see that both AMT and SIT achieve better performance over the baseline model on *Rest14*, *Laptop* and *Rest16* datasets. The results suggest that AMT sufficiently generalizes the target task by transferring knowledge from the auxiliary task. On the other hand, the performance of SIT suggests that this method can select and utilize transferable instances from the auxiliary task to enlarge the target task's dataset. Furthermore, we find that AMT+SIT significantly outperforms both AMT and SIT. We observe that this combined method takes advantage of the power of model transfer and instance transfer to fully exploit knowledge in auxiliary task.

We also compare the performance of AMT, SIT and AMT+SIT methods with several existing approaches, including non-transfer methods: LSTM & TD-LSTM [48], ATAE-LSTM [56], MemNet [51], RAM [8], MGAN [13], TNet [30], and transfer methods: MGAN [31], PRET+MULT [22] and TransCap [9]. The results are illustrated in Table 3. The performance of SIT is competitive with the contemporary state-of-the-art method, and the performance of AMT exceeds the state-of-the-art results on *Rest14* and *Rest16*. By combining AMT+SIT, we further surpass the result of both SIT and AMT, generating the new state-of-the-art result on these datasets.

Furthermore, we observed that the performance of our proposed methods largely depends on the task and domain similarity between the datasets in a dataset pair. For instance, the performance on *Rest14* is high when assisted by *Yelp*, then dropped when assisted by *Laptop*, and further lower when assisted by *Elec*. Based on this observation, we can infer that for all auxiliary datasets, the performance boost it can provide is as the following order: Cross Task only > Cross Domain only > Cross Task & Domain. In the following section, we will show that the combination of AMT and SIT can be both effective and robust to low task/domain similarity.

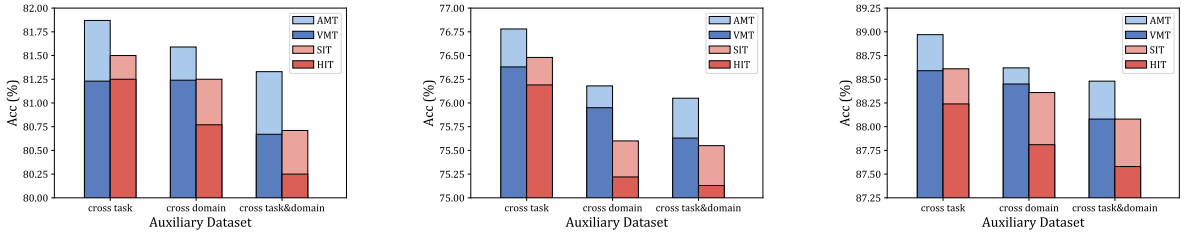
4.4 The Effectiveness of Soft transferring

In this section, we compare AMT and SIT with the respective hard transferring methods: Vanilla Model Transfer (VMT) and Hard Instance Transfer (HIT). The VMT method uses a hard parameter sharing method [5] which forces the parameters θ_A of the shared modules A to be equal in both auxiliary and target tasks:

$$\theta_A^{\mathcal{Y}} = \theta_A^{\mathcal{Z}}$$

The HIT method substitute the *sigmoid* function in SIT with a step function. The additional loss term for HIT is shown below:

$$\ell_{\mathcal{Z}\to\mathcal{Y}} := \sum_{x\in\mathcal{X}_Z} s\left(\mathcal{P}(\hat{y}) - \beta\right) \mathrm{CE}\left(\mathcal{P}(\hat{Y}), \hat{y}\right)$$



(a) The accuracy on Rest14 dataset. (b) The accuracy on Laptop dataset. (c) The accuracy on Rest16 dataset.

Figure 3: Improvement of AMT compared to VMT, and SIT compared to HIT on all dataset pairs.

where $s(p) = 1$ for $p > 0$, otherwise $s(p) = 0$.

We conduct the experiments on the three datasets, the results of the comparison on model transfer and instance transfer are shown in Figure 3. Results from the experiment suggest that AMT and SIT outperform VMT and HIT respectively with varied datasets used as the auxiliary task. Although we experience a performance increase when we replace VMT with AMT for model transfer or replace HIT with SIT for instance transfer, we find that training with different auxiliary datasets results in varying degrees of a performance boost. We provide details from this experiment.

Anchored Model Transfer vs. Vanilla Model Transfer Taking *Rest14* as the target dataset, AMT achieves better performance over VMT by gaining a performance boost of 0.79% and 0.82% when we adopt Cross Task and Cross Task & Domain dataset pair respectively. However, using the Cross Domain dataset pair, the performance boost of the AMT over the VMT is merely 0.43%. A similar situation occurs when using *Laptop* or *Rest16* as the target task.

We believe that in the VMT method, hard parameter sharing tend to integrate features of the two datasets into a shared model, allowing it to fit tasks with high similarity by safely sharing parameters between the auxiliary and target tasks. However, as for less relevant auxiliary tasks, it can be foreseen that the optimized model parameters for target and auxiliary tasks largely differ from each other, hence increasing the limitation brought by VMT. On the contrary, AMT is able to differentiate the parameters of the target and auxiliary task, while ensuring the two tasks to share the jointly learned knowledge. In conclusion, AMT is superior to VMT, and for less relevant auxiliary tasks, the boost brought by AMT is due to better model fitting, and can ultimately produce higher performance boost when replacing VMT to AMT.

Soft Instance Transfer vs. Hard Instance Transfer Then taking *Rest14* as the target dataset, SIT achieves better performance over HIT by gaining a performance boost of 0.57% and 0.59% when we adopt Cross Domain and Cross Task & Domain dataset pair respectively. However, when training with the Cross Task dataset pair, the performance boost of the SIT over the HIT is merely 0.31%. A similar situation occurs when *Laptop* or *Rest16* is used as target dataset.

HIT method uses static threshold and confidence-irrelevant weight and can discard instances even with confidence slightly lower than the threshold, which might result in a low performance when the target and auxiliary dataset have a dissimilar domain. Since Cross Domain dataset pairs have relatively low predict confidence, when using the HIT method, only a small proportion of auxiliary instances can be transferred, so the auxiliary dataset is not fully exploited. On the contrary, SIT manages to utilize the instances discarded by HIT and weight each instance according to its confidence, thus can utilize low-confidence instances to improve the model's performance in domains different from the target dataset.

5 CONCLUSION

Several works for aspect-level sentiment classification face sparsity and quality representations problems. To address these problems, in this paper we propose transfer learning approaches which aim at fully exploiting knowledge from cross-task or cross-domain to address sparsity and representation learning in ASC. Accordingly, we design two transfer learning methods AMT and SIT, which are both based on multi-task learning and are respectively used for model transfer and instance transfer for performance improvement in ASC. AMT allows other fine and related information to be learned from auxiliary tasks through an anchored parameter sharing layer, while SIT allows instances related to the domain of the ASC task to be transferred to enlarge ASC datasets. Experimental results show that AMT and SIT achieve competitive performance on benchmark datasets. We also compare the proposed methods with the corresponding hard transferring methods to validate the effectiveness of soft transferring, showing that our methods can achieve better performance in cross-task or cross-domain situations. Moreover, our proposed joint method AMT+SIT takes advantage of the model transferability of AMT and the instance transferability of SIT to boost performance significantly, achieving state-of-the-art performance on benchmark datasets. More importantly, the proposed methods AMT, SIT and AMT+SIT can be applied in other NLP applications, including named entity recognition (NER), and aspect extraction (AE). In the future, we intend to explore its application in those fields.

ACKNOWLEDGMENTS

This work is supported partly by the National Natural Science Foundation of China (No. 61772059, 61421003), by the Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), by State Key Laboratory of Software Development Environment (No. SKLSDE-2018ZX-17), by the Beijing S&T Committee (No. Z19110000 8619007) and by the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *ICML*. 214–223.

[2] Isabelle Augenstein, Sebastian Ruder, and Anders Søgaard. 2018. Multi-Task Learning of Pairwise Sequence Classification Tasks over Disparate Label Spaces. In *NAACL*. 1896–1906.

[3] Bart Bakker and Tom Heskes. 2003. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research* 4, May (2003), 83–99.

[4] Parminder Bhatia, Kristjan Arumae, and E Busra Celikkaya. 2019. Dynamic Transfer Learning for Named Entity Recognition. In *International Workshop on Health Intelligence*. 69–81.

[5] Rich Caruana. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *ICML*. 41–48.

[6] Hang Chang, Ju Han, Cheng Zhong, Antoine M Snijders, and Jian-Hua Mao. 2017. Unsupervised Transfer Learning via Multi-Scale Convolutional Sparse Coding for Biomedical Applications. *TPAMI* 40, 5 (2017), 1182–1194.

[7] Minmin Chen, Zhixiang Xu, Kilian Q Weinberger, and Fei Sha. 2012. Marginalized Denoising Autoencoders for Domain Adaptation. In *ICML*. 1627–1634.

[8] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In *EMNLP*. 452–461.

[9] Zhuang Chen and Tieyun Qian. 2019. Transfer Capsule Network for Aspect Level Sentiment Classification. In *ACL*. 547–556.

[10] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *SSST-8*. 103–111.

[11] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007. Boosting for transfer learning. In *ICML*. 193–200.

[12] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification. In *ACL*. 49–54.

[13] Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained Attention Network for Aspect-Level Sentiment Classification. In *EMNLP*. 3433–3442.

[14] Xing Fan, Emilio Monti, Lambert Mathias, and Markus Dreyer. 2017. Transfer Learning for Neural Semantic Parsing. In *Rep4NLP*. 48–56.

[15] George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. In *EMNLP*. 451–459.

[16] Yoav Freund and Robert E Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*. 23–37.

[17] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *ICML*. 1180–1189.

[18] Sheng Gao and Haizhou Li. 2011. A cross-domain adaptation method for sentiment classification using probabilistic latent analysis. In *CIKM*. 1047–1052.

[19] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.

[20] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *ICML*. 513–520.

[21] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective Attention Modeling for Aspect-Level Sentiment Classification. In *COLING*. 1121–1131.

[22] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting Document Knowledge for Aspect-level Sentiment Classification. In *ACL*. 579–585.

[23] Yulan He and Deyu Zhou. 2011. Self-training from labeled features for sentiment analysis. *Information Processing & Management* 47, 4 (2011), 606–616.

[24] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *ICASSP*. 7304–7308.

[25] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *ACL*. 151–160.

[26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[27] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *SemEval*. 437–442.

[28] Zeyang Lei, Yujiu Yang, Min Yang, Wei Zhao, Jun Guo, and Yi Liu. 2019. A Human-Like Semantic Cognition Network for Aspect-Level Sentiment Classification. In *AAAI*. 6650–6657.

[29] Na Li, Huizhen Hao, Qing Gu, Danru Wang, and Xiumian Hu. 2017. A transfer learning method for automatic identification of sandstone microscopic images. *Computers & Geosciences* 103 (2017), 111–121.

[30] Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation Networks for Target-Oriented Sentiment Classification. In *ACL*. 946–956.

[31] Zheng Li, Ying Wei, Yu Zhang, Xiang Zhang, and Xin Li. 2019. Exploiting Coarse-to-Fine Task Transfer for Aspect-Level Sentiment Classification. In *AAAI*. 4253–4260.

[32] Xiaobo Liu, Zhentao Liu, Guangjun Wang, Zhihua Cai, and Harry Zhang. 2017. Ensemble Transfer Learning Algorithm. *IEEE Access* 6 (2017), 2389–2396.

[33] Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit Discourse Relation Classification via Multi-Task Neural Networks. In *AAAI*. 2750–2756.

[34] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. In *ICML*. 97–105.

[35] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive Attention Networks for Aspect-Level Sentiment Classification. In *IJCAI*. 4068–4074.

[36] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. 43–52.

[37] David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *NAACL*. 152–159.

[38] Jan Niehues and Eunah Cho. 2017. Exploiting Linguistic Resources for Neural Machine Translation Using Multi-task Learning. In *WMT*. 80–89.

[39] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *CVPR*. 1717–1724.

[40] Sinno Jialin Pan and Qiang Yang. 2009. A Survey on Transfer Learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[41] David Pardoe and Peter Stone. 2010. Boosting for Regression Transferr. In *ICML*. 863–870.

[42] Nanyun Peng and Mark Dredze. 2016. Multi-task Multi-domain Representation Learning for Sequence Tagging. *CoRR* abs/1608.02689 (2016).

[43] Yong Peng, Shen Wang, and Bao-Liang Lu. 2013. Marginalized Denoising Autoencoder via Graph Regularization for Domain Adaptation. In *ICONIP*. 156–163.

[44] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. 1532–1543.

[45] Slav Petrov, Ryan Mcdonald Google, New York, and Ny. 2012. Overview of the 2012 Shared Task on Parsing the Web. (2012).

[46] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, et al. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *SemEval*. 19–30.

[47] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *SemEval*. 27–35.

[48] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for Target-Dependent Sentiment Classification. In *COLING*. 3298–3307.

[49] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *EMNLP*. 1422–1432.

[50] Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning Semantic Representations of Users and Products for Document Level Sentiment Classification. In *ACL-IJCNLP*. 1014–1023.

[51] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In *EMNLP*. 214–224.

[52] Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, and Jiebo Luo. 2019. Progressive Self-Supervised Attention Learning for Aspect-Level Sentiment Analysis. In *ACL*. 557–566.

[53] Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. 2017. Multitask Learning with Low-Level Auxiliary Tasks for Encoder-Decoder Based Speech Recognition. In *Interspeech 2017*. 3532–3536.

[54] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial Discriminative Domain Adaptation. In *CVPR*. 7167–7176.

[55] Bailin Wang and Wei Lu. 2018. Learning Latent Opinions for Aspect-level Sentiment Classification. In *AAAI*. 5537–5544.

[56] Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *EMNLP*. 606–615.

[57] Zhenghui Wang, Yanru Qu, Liheng Chen, Jian Shen, Weinan Zhang, Shaodian Zhang, Yimei Gao, Gen Gu, Ken Chen, and Yong Yu. 2018. Label-Aware Double Transfer Learning for Cross-Specialty Medical Named Entity Recognition. In *NAACL*. 1–15.

[58] Yi Yang and Jacob Eisenstein. 2014. Fast Easy Unsupervised Domain Adaptation with Marginalized Structured Dropout. In *ACL*. 538–544.

[59] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-Task Cross-Lingual Sequence Tagging from Scratch. *CoRR* abs/1603.06270 (2016).

[60] Yi Yao and Gianfranco Doretto. 2010. Boosting for transfer learning with multiple sources. In *CVPR*. 1855–1862.

[61] Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. 2016. Bi-Transferring Deep Neural Networks for Domain Adaptation. In *ACL*. 322–332.

[62] Yftah Ziser and Roi Reichart. 2017. Neural Structural Correspondence Learning for Domain Adaptation. In *CoNLL*. 400–410.