Cross-Domain Sentiment Classification with Attention-Assisted GAN

Yi-Fan Li, Yu Lin, Yang Gao and Latifur Khan
Department of Computer Science
The University of Texas at Dallas
Richardson, TX
{yli, yxl163430, yxg122530, lkhan}@utdallas.edu

2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI) | 978-1-6654-1621-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/COGMI52975.2021.00020

***Abstract*—In the problem setting of cross-domain sentiment classification, two different domains are introduced, and we refer to them as source and target domains respectively. In the source domain, sentiment labels to instances are available, while labels to instances in the target domain are not available. This problem is critical and practical, as in the real world, data in some domains (source) are abundant, while those in other domains (target) may become scarce. In this paper, we propose a cross-domain sentiment classification framework based on Generative Adversarial Networks (GANs) with the assistance of the attention mechanism, which aims to leverage the information available from the source domain to the target domain. Existing state-of-the-art methods mainly use multi-task learning to minimize the distance between the source and the target instances in a latent feature space. However, the projections may suffer as the deep model always tries to overfit the cross-domain adaptation task. We introduce a framework with multiple tasks, including adversarial example generation, cycle reconstruction, and cross-domain classification. Empirical evaluation and analysis on real-world datasets are being performed to validate the effectiveness of our proposed algorithm compared to state-of-the-art techniques.**

***Index Terms*—domain adaptation, sentiment classification, attention-assisted GAN**

I. Introduction

Deep learning has been performing extremely well in a wide range of machine learning tasks, and it is also one of the most exciting areas regarding recent progress within computer science community. Generally the advances of deep neural networks (DNNs) depend on a huge amount of labeled training data. These massive training data enable DNN to learn the patterns with their immense amount of parameters. However, labeling data is very expensive in real-world cases. In many cases, direct access to data labels is just not available. Therefore, knowing how to properly use knowledge from similar domains is becoming a popular method to solve in this category of problems. This type of approach is known as transfer learning.

Under a transfer learning setting, the sentiment classification problem assumes that data distribution between source and target domains is different but related. This assumption is very different from the traditional problem setting, where source and target datasets are from the same distribution. Specifically, if we conduct sentiment analysis on different domains, then it is important to notice that those keywords that best represent sentiment are completely different because it will lead to different data distributions.

Accordingly, traditional classification methods may not work under this problem setting. For instance, in *DVD* review domain, *watchable* may represent positive sentiments, whereas *pointless* may represent negative sentiments. However, when it comes to *Electronics* review domain, *glossy* may indicate positive sentiments and *slow* may indicate negative sentiments. This kind of discrepancy on sentiment distributions may lead to degraded performance of traditional classification algorithms.

Researchers have proposed different frameworks to solve the sentiment classification problem across multiple domains. Among various kinds of solutions, most of them try to project features from multiple domains into a common latent feature space for alignment. With the recent advances in deep learning, powerful tools, such as the Stacked Denoising Autoencoder (SDAE) [4] and Convolutional Neural Network (CNN) [11], have proven to be successful in tasks such as generating latent features in space using highly non-linear functions. However these methods only consider using a portion of information in transfer learning tasks as single task learning. Furthermore, they are highly dependent on manually identifying positive and negative pivots for alignment.

Recent advances in Generative Adversarial Networks (GANs) had shown their potential in generating discrete data [12], especially text sequences [26]. These state-of-the-art GANs normally used Reinforcement Learning (RL) type of algorithm to train the generator due to the nature that text generation is a discrete data generation process. However, in past research only consider this cross-domain adaptation as a single direction problem.

In this paper, the following problem settings are examined: **1)** Multiple domains: it refers to source data that may have a different distribution with regard to target data; **2)** Unsupervised training for target data: it means labels are only available to source data, whereas target data may not have any label; **3)** Sentiment classification: it means that we want to identify whether the sentiment of each data instance is positive or negative.

To address problem setting described above, a novel domain adaptation framework (CGAN) is proposed to align the differences across the different domains. The basic idea is to find good generator functions across domains, which help map data from the source domain to the target domain, such that distributions of generated data (from source) aligns with

978-1-6654-1621-4/21/$31.00 ©2021 IEEE
DOI 10.1109/CogMI52975.2021.00020

that from the distribution of data from target domain, vice versa. Subsequently, the data generated from the source data along with its labels can be used for various of different tasks, including sentiment classification for target domain as a unsupervised problem setting (regarding that there isn't any label available for the target domain).

To achieve our goal, we make the following contributions:

- GAN is used simultaneously across the source and target domain to minimize our designed loss, which aims to match the data distribution between generated data and real data for both domains.
- Attention mechanism is introduced to the discriminator in one domain, and it is incorporated with the generator in the other domain to help with improving the quality of generated text by the generator. To our best knowledge, this is the first attempt to handle the performance coordination between generator and discriminator in text data.
- Our proposed framework is compared to state-of-the-art methods on various commonly used benchmark Amazon Review dataset, which includes four categories: *DVD, Electronics, Books, and Kitchen*, and have shown effectiveness of our work.

II. Related Work

A. Domain Adaptation and Generative Adversarial Networks (GANs)

One of the fundamental assumptions of data mining, known as the "stationary distribution assumption," is that both the training and testing data are generated from the same domain, and thus they represent the same data distribution. The cross-domain sentiment classification problem, a transfer learning task, has attracted a lot of attention in recent years. In this task, many common techniques cannot be directly employed because training and test data are from different domains. The differences between domains can be considered from two aspects: 1. distinct number of features; 2. distinct feature distributions. Various types of approaches have been proposed to address this problem, and most of them try to learn a common feature representation for domain adaptation [2]. Li et al. [13] proposes a linear objective function to embed a latent feature space for both source and target domains for data streams. Recent research has proposed multi-task learning as a powerful tool to learn a joint latent representation for different tasks via either hard sharing or soft sharing [23]. However, these authors used related tasks to learn in a single domain setting rather than using multiple domains.

There has been another line of work to automatically generate good feature representation using deep learning. Chen et al. [4] proposes marginalized Stacked Denoising Autoencoder (mSDA), a fast and scalable way to learn the latent feature representation across domains. With regard to NLP applications, Yu et al. [25] applies a CNN, as well as two auxiliary tasks to induce sentence embeddings across domains. Within the deep learning direction, another exciting area is GANs. Goodfellow et al. [7] proposes a training methodology to generative models. Even though this network is originally designed for adapting the output continuously, lots of work have been done to generate structured sequences during the recent years [8], which makes it a fascinating area to conduct research.

B. Attentions on NLP

NLP has proven to be a very successful area in applying the attention mechanism. Since one of major problems in this domain is about sequence modeling, the attentions generated could provide more insights along the sequences [24]. Initially, attention has mainly been used in machine translation problems [1]. Later, there is good discussion on how to use text information such as caption to explain the focus on the images [22], [24]. Recently, hierarchical attention network is proposed by automatically capturing pivots and non-pivots in the sentence for sentiment classification [14].

Recently, it has been shown that machine translation models could achieve best in class performance by solely using an attention model [20]. Based on the transformer unit designed, the BERT model [5] and its extension work have been dominating the tasks in natural language processing area. Now, the self-attention (intra-attention) mechanism has been widely applied to learning task-independent sentence representations [15]. However, none of these state-of-the-art methods discusses the attention generated by discriminator in GAN, and none of them mentions how this information could be beneficial to domain adaptation problems.

III. Problem Formulation

Assume that we have texts from two domains, where X_s represents data from the source domain, and X_t represents data from the target domain respectively. Fore X_s, there are source texts $x_s^i \in X_s$ and their corresponding labels are $y^i \in Y$. Similarly, there are target texts $x_t^j \in X_t$, and they are the same label space as the source data $y^j \in Y$. Our problem setting is unsupervised learning regarding target domain, which means that we do not have the true label information $y^j \in Y$. The goal is to learn good generator functions $G_{fwd} : X_s \rightarrow X_t$ and $G_{bwd} : X_t \rightarrow X_s$, along with two discriminator functions D_s and D_t. Here, G_{fwd} and G_{bwd} are generators that try to generate good quality adversarial examples from one domain to the other. D_s and D_t are essentially classifiers to distinguish whether an instance is fake or not. For example, D_t tries to identify whether a sentence is from x_t directly or from generated sentences $G_{fwd}(x_s)$, and vice versa. More importantly, the key idea besides applying GAN-based learning structure, is that an attention map generated by D_t is used in the training process of generator G_{fwd} so that quality of generated text could be improved.

To sum up, we propose two types of optimization objectives to solve the cross-domain sentiment classification problem:

- Attention-guided adversarial loss (forward path and backward path in Fig. 1) aims to match the data distributions

X_s	Source domain
X_t	Target domain
G_{fwd}	Generator network from source domain to target domain
D_s	Discriminator network in source domain
G_{bwd}	Generator network from target domain to source domain
D_t	Discriminator network in target domain
T	Attention network
F	Classifier network

TABLE I: Frequently used symbols

between generated adversarial examples from one domain and real examples from the other domain.

- Cycle consistency loss (cycle consistency path in Fig. 1) aims to prevent the generator to generate contradict instances. Also this optimization objective helps the framework from overfitting neither forward generation nor backward generation process.

IV. Proposed Approach

A. Architecture

Our general idea for addressing this unsupervised cross-domain sentiment classification problem, is to find generators across domains with the assistance of attention mechanism. However, finding such good quality generators are challenging tasks. For this, we will use GAN between source and target domains along with attention mechanism. More specifically, attention weights are generated using outcomes of discriminators, and help the generator focus on those weakly generated examples (more details can be found in Sec. IV-B).

First, let's recap the general framework of GAN briefly. A typical training process of GAN is based on generator-discriminator pair. The generator trains based on whether it succeeds in fooling the discriminator. That is, similar to a zero-sum game, the generator always wants to create instances that can fool the discriminator. On the other hand, the discriminator tries to distinguish if an instance given is a generated one (fake) or sampled one (real). Previously, during this game, generators can only learn information from the generated instance itself, and the predicted results by the discriminator (fake/real). In a lot of cases, we observe that discriminators can converge much faster than generators, which is a big problem. We consider that the attention map from D_t could be crucial for the generator G_{fwd}, as it can save the computational power and allow G_{fwd} to concentrate on those locations, where help the discriminator to make decisions. In other words, the attention map generated by the discriminator would accelerate convergence speed of generator.

For the unsupervised learning setting, we first consider the forward task, which is from the source domain to the target domain. The backward task shares the same structure of networks but only have different inputs/outputs. The architecture of our proposed CGAN can been seen in Fig. 1. There are five major components in our proposed CGAN framework: two sets of generator-discriminator pair (G_{fwd} and D_t, G_{bwd} and D_s); one shared attention feedback component T.

In Fig. 1, three sub-tasks are shown: **I)** The forward path, which targets to adapt source data to target data by G_{fwd}, D_t, and T (see Fig. 1 (b)). **II)** The backward path, which targets to adapt target data to source data by G_{bwd}, D_s and T (see Fig. 1 (c)). **III)** The cycle consistency path, which targets to reconstruct source and target data respectively (see Fig. 1 (d)). Again, here we demonstrate the workflow of CGAN using forward path as an example. The backward path is exactly the same as forward path except the inputs and outputs.

Here, for simplicity, we denote $x_t^{i\prime} = G_{fwd}(x_s^i)$ as the output of generator, given x^i. Normally, the task for generator is to transform an instance (sentence) from the source domain ($x_s^i \in X_s$) to an instance (sentence) from the target domain ($x^{i\prime}_t \in \mathrm{X}_t$). Then, the discriminator involves in the training process. In this case, D_t tries to identify whether its inputs are generated fake sentence $x_t^{i\prime}$ from X_s, or a real sentence x_t^i directly got from the X_t. Additionally, we propose an module to obtain the attention embedding from $D_t(x_t^{i\prime}) \in [0, 1]$. The purpose of this attention map $A_{x_t^i}$ is to highlight the focusing words of D_t, which is basically those words that provide high gradient in the discriminator. Then, we pass this attention embedding $A_{x_t^i}$ to create an weight map, so that it can be incorporated to the G_{fwd}, and we hereafter refer this term as $M_{x_t^i}$.

Based on the demonstrations above, now we can formalize the training process of the generator G_{fwd}. The input of this generator is the concatenation of $M_{x_t^i}$ and x_s^i, and we denote this concatenation as $x^{i\prime}_s$. We use random initialization technique here to the attention map since we do not have this part of knowledge before hand. Combining all before mentioned modules together, the translation process of generator G_{fwd} can be formulated as:

$$x^{i\prime}{}_{t(k+1)} = G_{fwd}(x_s^i \oplus A(D_t(x^{i\prime}{}_{t(k)})); \theta) \tag{1}$$

where $k \in (0, 1, 2, ...)$ represents the number of iterations and θ is the denotes the index of iteration and θ is the parameter of G_s.

B. Network Modules

In this section, we discuss the attention mechanism in detail and how to incorporate it with original inputs.

The backbone of GAN in our proposed framework follows [28]. That is, we use the CNN architecture [11] for sentence encoding, and the LSTM architecture for text generation. Recall that an attention map $A_{x_t^i}$ is generated from D_t for every instance $x_t^i \in X_t$. A straightforward way to use this attention information is to consider the *post hoc attention* mechanism. In short, the basic idea of this attention mechanism is to get more information from the forward activations and backward gradient from D_t.

The network can be formulated as $D = \{l_0, l_1, \ldots, l_m\}$ where l_i denotes i-th convolution layer in the network, and $Act_D = \{a_1, a_2, \ldots, a_m\}$ is the set of activation map of corresponding layer. This kind of attention map is sensitive to layer selection; different layer selection leads to different attention map [16]. More specifically, if t is the chosen layer, the attention map can be described as:

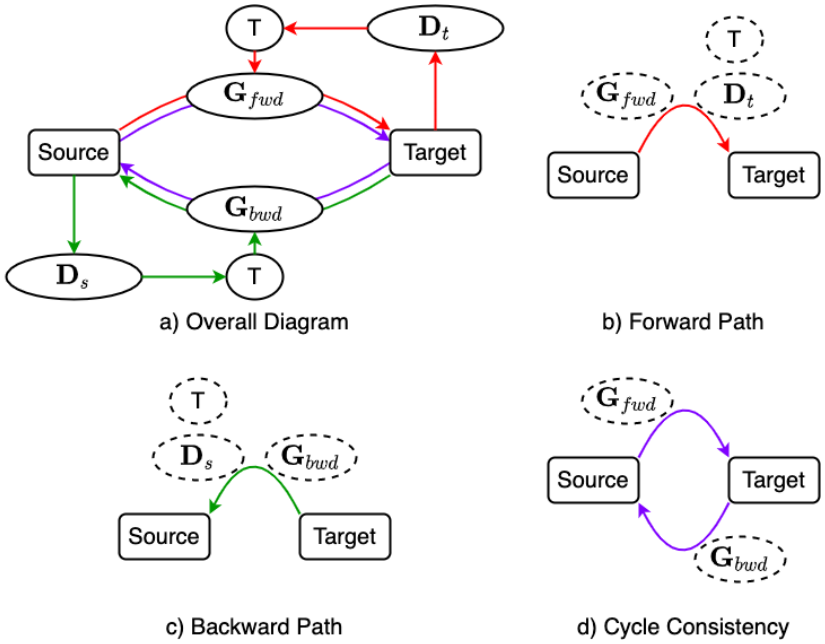


Fig. 1: Architecture of our proposed CGAN framework (Red line shows forward path; Green line shows backward path, and; Purple line shows cycle consistency)

$$M_x = g(\frac{1}{c}\sum_{i=1}^{c}|a_{t,i}|) \tag{2}$$

where c is the number of channels in t-th layer and $g(\cdot)$ applies the min-max normalization.

After getting M, we follow the process described in [21] to concatenate the attention map with its corresponding input. By applying the Residual Hadamard Production (RHP) process [17], we avoid the possibility on attention masks to break the good property of the raw input, as the element-wise production with the attention (between zero and one) may lead to degrade of the word embedding value. This RHP can be formulated as:

$$x'_s = (g(M_x;\phi)+1)\times x_s \tag{3}$$

where $g(;\phi)$ is a transfer function that up-sample the attention map to the shape of original input.

C. Overall Training Loss

Now we can take a look at the overall loss function for our CGAN framework. The adversarial loss of the generator G and its discriminator D can be expressed as:

$$\begin{aligned}L_{GAN}(G_{fwd}, D_t) = \mathbb{E}_{x_t\sim X_t}[\log D_t(x_t)] \\ + \mathbb{E}_{x_s\sim X_s}[\log(1-D_t(G_{fwd}(x_s\oplus M_x)))]\end{aligned} \tag{4}$$

The above function is the adversarial loss of vanilla GAN. Similar to a zero-sum game, G_{fwd} aims to minimize this objective while an adversary D_t tries to maximize it. In practice, this function is very hard to train. Therefore, least-squares loss are incorporated to stabilize the training process. Then the adversarial loss can be written as:

$$\begin{aligned}L_{GAN}(G_{fwd}, D_t) = \mathbb{E}_{x_t\sim X_t}[(D_t(x_t)-1)^2] \\ + \mathbb{E}_{x_s\sim X_s}[(G_{fwd}(x_s\oplus M_x))^2]\end{aligned} \tag{5}$$

According to [26], if we only apply the adversarial loss, the quality of text translation would not be of good quality. A typical way in computer vision applications is that we can add traditional loss, such as L1 or L2 regularization for unsupervised learning setting [29]. We further apply this idea by adding a pair of generator and discriminator and enforcing cycle consistency. Assume the generator G_{fwd} simulates the map function $G: X_s \rightarrow X_t$ and discriminator D_t are trying to distinguish between $G(x_s)$ and x_t, the objective of this GAN component is $L_{GAN}(G_{fwd}, D_t)$. On the other hand, for the opposite tasks, the generator G_{bwd} and discriminator D_s is doing the same task in an opposite direction, their loss function is $L_{GAN}(G_{bwd}, D_s)$. Here, the cycle consistency is used since in unsupervised learning, we are lacking the information about paired data. Therefore the cycle consistency enforced a close reconstruction about the instance itself, and it assumes that if a text x from domain X_s has be translated to a fake text $\hat{x_t}$ in domain X_t, we should get the same text x by applying $G_{bwd}: X_t \rightarrow X_s$. This process can be concluded using the following equations:

$$L_{cyc}(G_{fwd}, G_{bwd}) = \mathbb{E}_{x_s \sim X_s}[\|G_{bwd}(G_{fwd}(x'_s)) - x_s\|_1] + \mathbb{E}_{x_t \sim X_t}[\|G_{fwd}(G_{bwd}(x'_t)) - x_t\|_1] \quad (6)$$

The final objective function for the unsupervised translation is:

$$G^*_{fwd}, G^*_{bwd} = \arg \min_{G_{fwd}, G_{bwd}} \max_{D_s, D_t} (L_{GAN}(G_{fwd}, D_t) + L_{GAN}(G_{bwd}, D_s) + \lambda L_{cyc}(G_{fwd}, G_{bwd})) \quad (7)$$

Once generators and discriminators are properly trained, we use the last convolutional layer in the discriminator as the feature embeddings for $x^{i\prime}_t$ and apply its corresponding sentiment label from the source domain for training the classification task. This classification task can be conducted by a fully connected 3-layers neural network F using cross entropy loss for binary classification (positive/negative sentiment).

V. Experiment

A. Datasets and Implementations

We have conducted the experiments on the Amazon reviews dataset [3], which has been widely used for cross-domain sentiment classification. This dataset contains reviews from four products/domains: Books (B), DVD (D), Electronics (E), and Kitchen (K). There are 6,000 labeled reviews for each domain with 3,000 positive reviews (higher than 3 stars) and 3,000 negative reviews (lower than 3 stars). Following the same experiment setting as [14], we use 5,600 instances as training set and 400 instances as testing.

By following [19], we have constructed cross-domain sentiment classification tasks like A→B, where A corresponds to the source domain and B denotes the target domain. We compared our methods using different word embeddings to validate its effectiveness. First, we use Word2Vec embedding to align with the original baselines settings in previous research, and demonstrate the suporior performance of our proposed framework. Second, we use BERT embedding to further demonstrate the effectiveness of our model, even build on the top of BERT embeddings. Therefore, we pre-process each dataset instance via pretrained Word2Vec [18] and BERT model [5] to produce feature vectors with dimensions of 300 and 768 respectively in Spacy [10]. Then those vectors are used to the later experiments.

Recall that the backbone of our proposed CGAN is a standard CNN-LSTM autoencoder. According to [28], pre-training is very important to find good initializations for the generator (LSTM). Therefore, we follows the same pre-train protocal on the reviews from clothing category and sports category [9]. More specifically, we randomly select 250k reviews from clothing category and 250k reviews from the sports category. In total, the model is pre-trained on a dataset with 500k reviews. This pre-training process is conducted for Word2Vec embeddings and BERT embeddings separately. We use the parameters from this pretrained model as initializations to our CGAN, and conduct further training on the four categories mentioned before (Books, DVD, Electronics, and Kitchen)

Our proposed framework is implemented on *Python* 3.7.3 and *PyTorch* 1.2.0. All experiments are conducted on a workstation with Nvidia Quadro 8000 GPU. For unsupervised training phase, we set the weight factor of cycle consistency λ to be 10. The maximum length of review is set to be 50, and the size of minibatch is set to 128. We use Adam optimizer, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for generators and discriminators respectively. Learning rates for both generators and discriminators are 0.0005. The model is trained for 50 epochs.

B. Benchmark Method

As we discussed before, we apply two pretrained models to generate the word embedding to validate our proposed framework. We feed generated word vectors into our baseline methods. Detailed descriptions regarding baselines are as follows:

- DAmSDA [6]: this method applies mSDA [4], and further uses DANN to generate feature projections in the latent feature space. Embeddings from 5 layers are concatenated together to form the final representation.
- CNN-AUX [25]: this method applies CNN and two auxiliary tasks to achieve the latent feature embedding.
- IATN [27]: this method implements an interactive attention transfer mechanism, which create better feature embedding by incorporating information of both sentences and aspects.
- HATN [14]: this method uses hierarchical position encoding to locate positional embedding and makes use of domain-shared representation.
- BERT [5]: this method is fine tuning vanilla BERT by the source domain labeled data.
- BERT-HATN: this is HATN applications based on BERT word embedding.

C. Results

For this research, we focus on the sentiment classification results. We first demonstrate the classification accuracy regarding different frameworks applying the Amazon Review dataset. The experiment results using Word2Vec embedding are shown in Tab. II, and those using BERT embedding are shown in Tab. III. Since for each source domain we have three other domains, in total, there are 12 different combinations of transfer tasks. For instance, for *Books* domain, we have four transfer tasks *Books→DVD*, *Books→Electronics*, and *Books→Kitchen*.

From the result table, we can see that our proposed CGAN consistently over-performs all other competing baseline methods in most of the tasks, and our method get better average performance. Since for the source-only method, there isn't any domain adaptation at all, the result reflects our expectation that it would perform poorly on average with 75.20% accuracy. For the CNN-aux method, our proposed framework surpasses its performance by 5.04% due to the fact that the combination of the attention mechanism and our proposed cross-domain GANs based tasks help learn better transformation functions and cross-domain representations. Compared to the HATN, our designed attention map, which is generated by the discriminator

Source	Target	Source-only	DAmSDA	CNN-aux	IATN	HATN	CGAN
Books	DVD	0.805	0.861	0.844	0.868	0.870	**0.883**
	Electronics	0.716	0.790	0.806	**0.865**	0.857	0.863
	Kitchen	0.736	0.810	0.833	0.859	0.870	**0.879**
DVD	Books	0.764	0.851	0.830	0.870	0.877	**0.892**
	Electronics	0.731	0.761	0.803	**0.869**	0.863	0.858
	Kitchen	0.734	0.826	0.816	0.858	0.874	**0.876**
Electronics	Books	0.688	0.799	0.773	0.818	0.840	**0.861**
	DVD	0.720	0.826	0.790	0.841	0.843	**0.867**
	Kitchen	0.846	0.858	0.871	0.887	0.900	**0.903**
Kitchen	Books	0.715	0.805	0.784	0.847	0.848	**0.852**
	DVD	0.733	0.821	0.790	0.844	0.847	**0.860**
	Electronics	0.831	0.880	0.867	0.876	0.893	**0.903**
Average		0.752	0.824	0.817	0.858	0.865	**0.874**

TABLE II: Experiment results on different combinations of transfer learning tasks with **Word2Vec** embedding (accuracy)

Source	Target	BERT	BERT-HATN	BERT-CGAN
Books	DVD	0.889	0.893	**0.896**
	Electronics	0.861	0.872	**0.878**
	Kitchen	0.890	0.894	**0.905**
DVD	Books	0.894	0.898	**0.907**
	Electronics	0.865	0.869	**0.873**
	Kitchen	0.875	**0.875**	0.874
Electronics	Books	0.865	0.871	**0.872**
	DVD	0.879	**0.888**	0.887
	Kitchen	0.916	0.920	**0.929**
Kitchen	Books	0.875	0.878	**0.879**
	DVD	0.873	0.878	**0.882**
	Electronics	0.904	0.903	**0.908**
Average		0.882	0.887	**0.891**

TABLE III: Experiment results on different combinations of transfer learning tasks with **BERT** embedding (accuracy)

provides back to the generator, helps it to better adapt to the target domain. From the results, we can see that our proposed multi-task learning framework can contribute to the better performance in learning good representations. Our proposed attention mechanism automatically select the direction with higher gradient in the discriminator, and send it back to the generator to help it for better generation. The overwhelming performance of our model compared to HATN proves the validity of our designed cross-domain reconstruction tasks.

We further validate the effectiveness of our proposed CGAN framework by utilizing BERT word embedding, and the results are shown in Tab. III. We compare it with the fine-tuned version of BERT. Also, HATN, which is the best baseline in Tab. II is selected to train again with word-level embedding generated by BERT. From this table, we can see that our framework consistently improve the performance of BERT model for cross-domain sentiment classification problems. Such improvement can also be seen when comparing BERT-HATN with BERT-CGAN in most cases. For instance, our model outperforms the BERT-HATN model by 0.89% on the $DVD \rightarrow Books$ task.

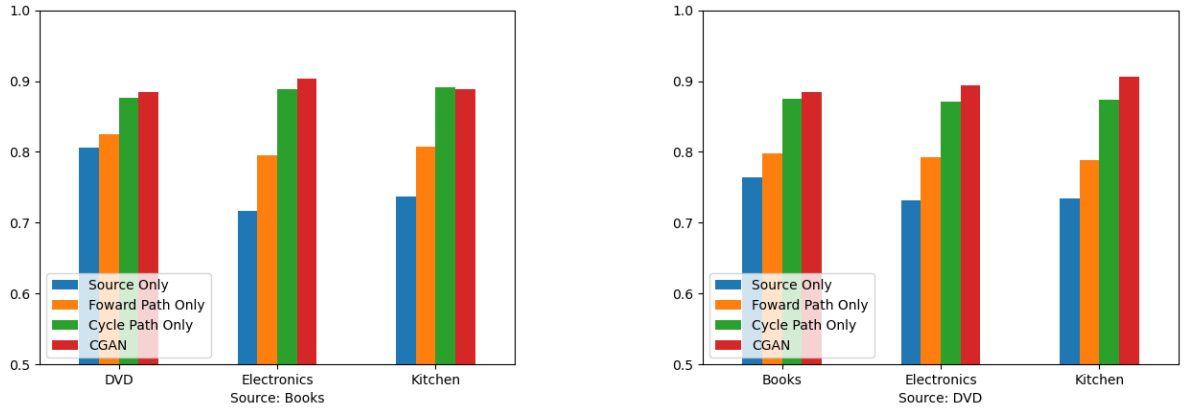
Based on the above experiment results, we can conclude that our framework is effective on finding proper generation functions for domain adaptation problems.

D. Comparisons of Domain Adaptation Tasks

We designed five different scenarios to discuss contributions of different tasks in our CGAN framework. The source-only methods, as we presented in previous section, uses pure source data to train the classifier following the settings of [14], and applies the trained classifier directly to the target data. Meanwhile, for the evaluation of forward path task, we only apply modules and tasks described by **b)** in Fig. 1 for cross-domain adaptation, and a typical classification module (fc layer). Regarding the cycle task only evaluation, we used cycle reconstruction tasks **d)** in Fig. 1 in addition to the classification tasks, which is basically CGAN_v in Table II.

From Fig.2, we could see that using GAN for forward path along can improve the classification performance by a significant margin, due to the powerful GAN mechanism. However, compared to its competitors, there is room to get a better cross-domain representation. For example, since only using the forward path may encourage the generator to overfit the target domain, and lose its semantic information, which may downgrade the model performance for identifying the correct sentiments. This assumption is proved by our observation by comparing the model performance between training set and testing set.

Also, we can see that the cycle reconstruction task provides significant contributions as well in our framework. This mechanism is extremely important here, since the data that we used in this paper is unpaired. If we consider the forward and backward generators along without the feedback of attention maps, the cycle consistency loss helps the model to align the sentiments across across different domains. This is because



(a) (b)

Fig. 2: Comparisons of scenarios using different set of transfer learning tasks (accuracy). Each subplot corresponds to results for tests from a certain source domain, and labels on x-axis are target domains. Here we demonstrate the results from two domains (Books & DVD). Results from other domains show similar performance.

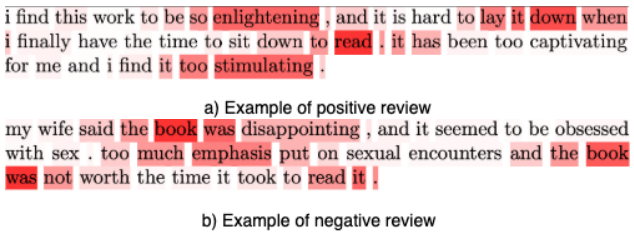


Fig. 3: Attention maps from D_t to G_{fwd} at epoch 20 on task: Books $\rightarrow$ DVD

by minimizing the loss of this task, it can help align the embedding distributions from source and target domains together. Therefore, the prediction accuracy rises dramatically when the cycle reconstruction loss is included.

Furthermore, we could see that the cycle reconstruction task overall prevent the forward only path from overfitting the training data. In the overall loss function, the self reconstruction loss performs as the regularization part, which force the framework to learn a good embedding, which ensures the good quality of reconstruction for source and target data respectively.

E. Attention Maps

Fig. 3 shows examples of attention map from D_t to G_{fwd} at epoch 20 on the task of $Books \rightarrow DVD$. Example a) shows a positive review whereas example b) shows a negative review. From these two examples, we can see that the attention generated from D_t tracks the differences between *Books* domain and *DVD* domain quite decent. For instance, in both cases, the word "read" is assigned with high attention. This result makes a lot of sense, as "read" normally related to books, other than DVD. Therefore, distribution differences between source and target domain are clearly provided by the attention mechanism generated from D_t, which is a great guidance for G_{fwd}.

Also, there are some attentions that doesn't quite make sense as well. For example, in the positive review, "stimulating" could exists in both *Books* domain and *DVD* domain. Therefore, this attention doesn't really provide G_{fwd} with a lot of good information. Nevertheless, the attention mechanism has excluded a lot of words in the sentence with low attention so that the generator can focus on some meaningful words, which is a good thing.

VI. Conclusion & Future Work

In this paper, we introduce a GAN based multi-task learning framework to address challenges in the cross-domain sentiment classification task. The unsupervised learning problem setting regarding the target domain is very difficult, and our framework have proved its superior performance by competing with state-of-the-art methods on even more difficult tasks. Especially, existing frameworks lack the additional guidance to cross-domain tasks on how to generate joint distributed features in a latent space. Our proposed framework overcomes this issue by using GAN based tasks and incorporating the attention map from discriminator to help the convergence of generator.

In the future, we plan to extend our work on even harder problem settings, such as source and target datasets that are in different languages, where the domain differences are greater than what we present in this paper. Additionally, we plan to investigate the importance of each module in the multi-task learning framework and find an adaptive way to automatically learn the weight for each loss function in order to improve the classification accuracy.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 129–136, 2010.

[3] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447, 2007.

[4] Minmin Chen, Zhixiang Xu, Kilian Q Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1627–1634. Omnipress, 2012.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[7] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

[8] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[9] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.

[10] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[11] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

[12] Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

[13] Yi-Fan Li, Yang Gao, Gbadebo Ayoade, Hemeng Tao, Latifur Khan, and Bhavani Thuraisingham. Multistream classification for cyber threat data with heterogeneous feature space. In *The World Wide Web Conference*, pages 2992–2998. ACM, 2019.

[14] Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. Hierarchical attention transfer network for cross-domain sentiment classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[16] Xiaoguang Mei, Erting Pan, Yong Ma, Xiaobing Dai, Jun Huang, Fan Fan, Qinglei Du, Hong Zheng, and Jiayi Ma. Spectral-spatial attention networks for hyperspectral image classification. *Remote Sensing*, 11(8):963, 2019.

[17] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 3693–3703, 2018.

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[19] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM, 2010.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[21] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pages 3156–3164, 2017.

[22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[23] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016.

[24] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[25] Jianfei Yu and Jing Jiang. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 236–246, 2016.

[26] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[27] Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. Interactive attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5773–5780, 2019.

[28] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4006–4015. JMLR. org, 2017.

[29] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.