# Hierarchical Attention Transfer Network for Cross-domain Sentiment Classification

**Zheng Li, Ying Wei, Yu Zhang, Qiang Yang**
Hong Kong University of Science and Technology, Hong Kong
zlict@cse.ust.hk, yweiad@gmail.com, yu.zhang.ust@gmail.com, qyang@cse.ust.hk

## Abstract

Cross-domain sentiment classification aims to leverage useful information in a source domain to help do sentiment classification in a target domain that has no or little supervised information. Existing cross-domain sentiment classification methods cannot automatically capture *non-pivots*, i.e., the domain-specific sentiment words, and *pivots*, i.e., the domain-shared sentiment words, simultaneously. In order to solve this problem, we propose a Hierarchical Attention Transfer Network (HATN) for cross-domain sentiment classification. The proposed HATN provides a hierarchical attention transfer mechanism which can transfer attentions for emotions across domains by automatically capturing pivots and non-pivots. Besides, the hierarchy of the attention mechanism mirrors the hierarchical structure of documents, which can help locate the pivots and non-pivots better. The proposed HATN consists of two hierarchical attention networks, with one named P-net aiming to find the pivots and the other named NP-net aligning the non-pivots by using the pivots as a bridge. Specifically, P-net firstly conducts individual attention learning to provide positive and negative pivots for NP-net. Then, P-net and NP-net conduct joint attention learning such that the HATN can simultaneously capture pivots and non-pivots and realize transferring attentions for emotions across domains. Experiments on the Amazon review dataset demonstrate the effectiveness of HATN.

## Introduction

Users usually express opinions about products or services on social media or review sites. It is helpful to correctly understand their emotional tendency. Sentiment classification, which aims to automatically determine the overall sentiment polarity (e.g., positive or negative) of a document, has raised continuous attentions over the past decades (Pang, Lee, and Vaithyanathan 2002; Hu and Liu 2004; Pang and Lee 2008; Liu 2012). Supervised learning algorithms that require labeled data have been successfully explored to build sentiment classifiers for a specific domain (Wang and Manning 2012; Socher et al. 2013; Tang et al. 2015). However, there exists insufficient labeled data in a target domain of interest, where labeling data may be time-consuming and expensive. Cross-domain sentiment classification, which borrows knowledge from related source domains with abundant labeled data to improve the target domain, has become a promising direction. However, the expression of users' emotions usually varies across domains. For example, in the *Books* domain, words *readable* and *thoughtful* are used to express positive sentiment, whereas *insipid* and *plotless* often indicate negative sentiment. On the other hand, in the *Electronics* domain, *rubbery* and *glossy* express positive sentiment, whereas words *fuzzy* and *blurry* usually express negative sentiment. Due to the domain discrepancy, a sentiment classifier trained in a source domain may not work well when directly applied to a target domain.

To address the problem, researchers have proposed various methods for cross-domain sentiment classification. Blitzer *et al.* (Blitzer, Dredze, and Pereira 2007) proposed a Structural Correspondence Learning (SCL) method which utilizes multiple pivot prediction tasks to infer the correlation between pivots and non-pivots. Pan *et al.* (Pan et al. 2010) proposed the Spectral Feature Alignment (SFA) to find an alignment between pivots and non-pivots by using the cooccurrence between them. However, these methods need to manually select pivots and they are based on discrete feature representations such as bag-of-words with linear classifiers. Recently, deep neural models are explored to automatically produce superior feature representations for cross-domain sentiment classification. Stacked Denoising Auto-encoders (SDA) have been successfully adopted to learn hidden representations shared across domains (Glorot, Bordes, and Bengio 2011; Chen et al. 2012). Yu and Jiang (Yu and Jiang 2016) leveraged two auxiliary tasks to learn sentence embeddings with a Convolutional Neural Network (CNN) (Kim 2014) which works well across domains, while they still rely on manually identifying positive and negative pivots. Ganin *et al.* (Ganin and Lempitsky 2015; Ganin et al. 2016) proposed the Domain-Adversarial training of Neural Networks (DANN) which first introduces a domain classifier incapable of discriminating representations from a source or a target domain by reversing the gradient direction of the neural network. In order to improve the interpretability of deep models, Li *et al.* (Li et al. 2017) proposed an Adversarial Memory Network (AMN) to automatically identity the pivots by using the attention mechanism and adversarial training. Nevertheless, AMN only focuses on word-level attention and ignores the hierarchical structure of documents, which may not accurately capture

pivots in long documents. Besides, it cannot automatically capture and exploit non-pivots, which may result in the degraded performance when source and target domains only have few overlapping pivot features.

To simultaneously harness the collective power of pivots and non-pivots and interpret what to transfer, we introduce a Hierarchical Attention Transfer Network (HATN) for cross-domain sentiment classification. The proposed HATN provides a hierarchical attention transfer mechanism, which can automatically transfer attentions for emotions in both word and sentence levels across domains to reduce the domain discrepancy and provide a better interpretability of what to transfer. Specifically, our framework consists of two hierarchical attention networks named P-net and NP-net, where P-net aims to focus on pivots and NP-net is used to identify non-pivots by using the pivots as a bridge. Firstly, P-net conducts individual attention learning to select positive and negative pivots for NP-net. Then, P-net and NP-net conduct joint attention learning such that the HATN can simultaneously identify the pivots and non-pivots and achieve transferring attentions for emotions across domains.

Our contributions are summarized as follows:

- We propose a hierarchical attention transfer mechanism, which can transfer attentions for emotions across domains by automatically capturing the pivots and non-pivots simultaneously. Besides, it can tell what to transfer in the hierarchical attention, which makes the representations shared by domains more interpretable.
- Empirically the proposed HATN method can significantly outperform the state-of-the-art methods.

## Related Work

**Domain Adaptation** Domain adaptation tasks such as cross-domain sentiment classification have raised much attention in recent years. One line of work focuses on inducing a low-dimensional feature representation shared across domains based on the cooccurrence between pivots and non-pivots. Unfortunately, they highly rely on manually selecting pivots based on term frequencies on both domains (Blitzer, McDonald, and Pereira 2006), mutual information between features and labels of a source domain (Blitzer, Dredze, and Pereira 2007), mutual information between features and domains (Pan et al. 2010), and weighted log-likelihood ratio (Yu and Jiang 2016). These pivot selection methods are very tedious and may not identify the pivots accurately.

Recently, deep learning methods form another line of work to automatically produce superior feature representations for cross-domain sentiment classification. SDA (Glorot, Bordes, and Bengio 2011) is proposed to learn to discover intermediate representations shared across domains. In order to improve the speed and scalability of SDA for high-dimensional data, Chen *et al.* (Chen et al. 2012) proposed a Marginalized Stacked Denoising Autoencoder (mSDA). Yu *et al.* (Yu and Jiang 2016) used two auxiliary tasks to help induce sentence embeddings with CNN across domains. Ganin *et al.* (Ganin and Lempitsky 2015; Ganin et al. 2016) proposed the DANN, which leverages the domain adversarial training method to make the neural network produce representations confusing a domain classifier. In order to improve the interpretability of deep models, Li *et al.* (Li et al. 2017) incorporated memory networks into DANN to automatically identify the pivots.

**Attention Mechanism in NLP** The attention mechanism has been successfully exploited in various NLP tasks such as machine translation (Bahdanau, Cho, and Bengio 2014), question answering (Sukhbaatar et al. 2015), document classification (Yang et al. 2016) and sentiment analysis (Tang, Qin, and Liu 2016). The intuition behind the attention mechanism is that each low-level position contributes a different importance for the high-level representation. Moreover, the hierarchical attention mechanism has been proved to be better than the word-level attention in various document-based tasks (Yang et al. 2016; **?**; Haoran Huang 2017), which mirrors the hierarchical structure of documents in order to extract more powerful features.

## Hierarchical Attention Transfer Network

In this section, we introduce the proposed HATN model for cross-domain sentiment classification. We first present the problem definition and notations, followed by an overview of the model. Then we detail the model with all components.

### Problem Definition and Notations

We are given two domains $D_s$ and $D_t$ which denote a source domain and a target domain, respectively. Suppose that we have a set of labeled data $\mathbf{X}_s^l = \{\mathbf{x}_s^i\}_{i=1}^{N_s^l}$ and $\{y_s^i\}_{i=1}^{N_s^l}$ as well as some unlabeled data $\mathbf{X}_s^u = \{\mathbf{x}_s^i\}_{i=N_s^l+1}^{N_s}$ in the source domain $D_s$, where $\mathbf{X}_s = \mathbf{X}_s^l \bigcup \mathbf{X}_s^u$. Besides, a set of unlabeled data $\mathbf{X}_t = \{\mathbf{x}_t^j\}_{j=1}^{N_t}$ is available in the target domain $D_t$. The goal of cross-domain sentiment classification is to train a robust classifier on labeled samples in the source domain $\mathbf{X}_s^l$ and adapt it to predict the sentiment polarity of unlabeled examples $\mathbf{X}_t$ in the target domain, which is also widely known as unsupervised domain adaptation.

### An Overview of the HATN Model

In this section, we present an overview of the proposed HATN model for cross-domain sentiment classification.

The goal of HATN is to transfer attentions for emotions across domains, i.e, automatically capturing the pivots as well as non-pivots. Therefore, we design two hierarchical attention networks with different attentions for the target. As illustrated in Figure 1, the first network is named P-net, which aims to identify the pivots shared by the source and target domains such as *great* in a sample $\mathbf{x}$. The second network is named NP-net which is used to capture the non-pivots across domains such as *readable* in a transformed sample $g(\mathbf{x})$, generated by hiding all the pivots like the positive pivots *great* in the sample $\mathbf{x}$. We realize the 'hide' action by replacing the pivots with padding words.

In order to demonstrate the effects of each network clearly, we describe the input, objective and motivation for the P-net and NP-net, respectively.

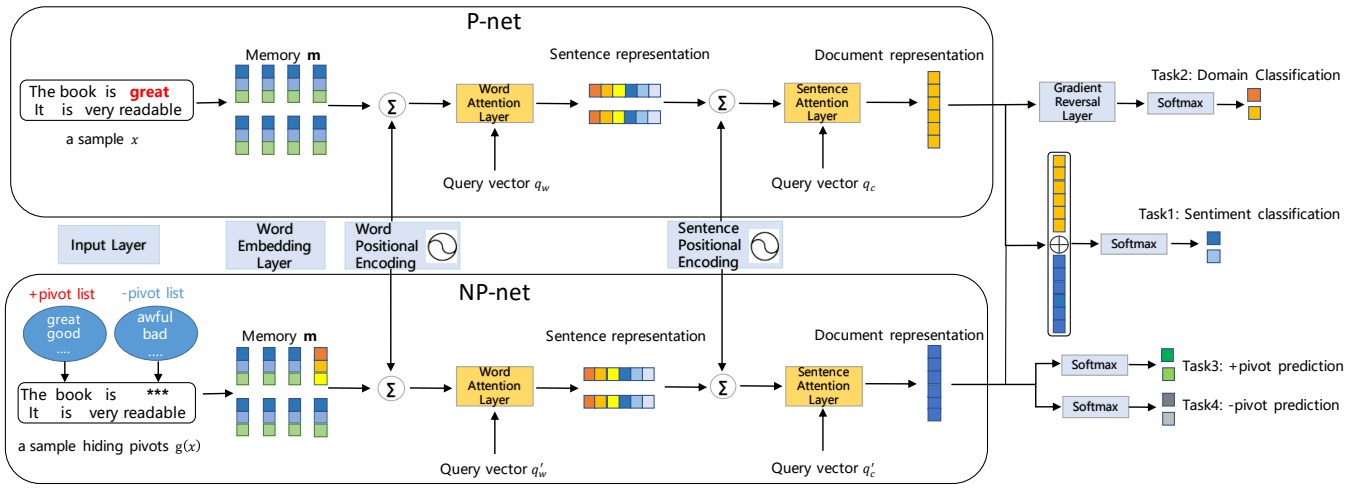


Figure 1: The framework of the HATN model.

**P-net** aims to capture the pivots, which have two attributes: (1) They are important sentiment words for sentiment classification. (2) They are shared by both domains. In order to achieve this goal, the labeled data $\mathbf{X}_s^l$ in the source domain is fed into the P-net for sentiment classification and in the meanwhile, all the data $\mathbf{X}_s$ and $\mathbf{X}_t$ in both domains are fed into the P-net for domain classification that aims to predict the domain label of the samples, i.e, coming from the source or target domain. Here we perform the domain classification based adversarial training by the Gradient Reversal Layer (GRL) (Ganin et al. 2016) such that make the domain classifier indiscriminative between the representations from the source and target domains. In this way, it guarantees representations from the P-net are both domain-shared and useful for sentiment classification, and can identity the pivots with the attention mechanism.

**NP-net** aims to capture the non-pivots with two characteristics: (1) They are the useful sentiment words for sentiment classification. (2) They are domain-specific words. To reach the goal, the transformed labeled data $g(\mathbf{X}_s^l)$ in the source domain $D_s$, generated by hiding all the pivots identified by the P-net, is fed into the NP-net for sentiment classification. And at the same time, all transformed data $g(\mathbf{X}_s)$ and $g(\mathbf{X}_t)$ in both domains $D_s$ and $D_t$, generated by the same way, is fed to NP-net for +(positive)/-(negative) pivot predictions. The two tasks aim to predict whether an original sample $\mathbf{x}$ contains positive or negative pivots based on the transformed sample $g(\mathbf{x})$. The transformed sample $g(\mathbf{x})$ has two labels, a label $z^+$ indicating whether $\mathbf{x}$ contains at least one positive pivot and a label $z^-$ indicating whether $\mathbf{x}$ contains at least one negative pivot. The intuition behind is that positive non-pivots tend to co-occur with positive pivots and negative non-pivots tend to co-occur with negative pivots. In this way, the NP-net can discover domain-specific features with the pivots as a bridge and capture the non-pivots that are expected to correlate closely to the pivots with the attention mechanism.

**Training Process:** Note that the NP-net needs the positive and negative pivots as a bridge across domains. Different from traditional methods that need to manually select pivots, the P-net possesses the ability of automatically finding the pivots. Therefore, our training process consists of two stages:

- Individual Attention Learning: The P-net is trained for cross-domain sentiment classification. We use the best parameters for P-net with early stopping on the validation set, and then select a word with the highest word attention in the sentence with the highest sentence attention in each positive review of the source domain as a positive pivot. The negative pivots are obtained in a similar way.
- Joint Attention Learning: The P-net and NP-net are jointly trained for cross-domain sentiment classification. The labeled data $\mathbf{X}_s^l$ and its transformed data $g(\mathbf{X}_s^l)$ in the source domain $D_s$ are simultaneously fed into P-net and NP-net respectively and their representations are concatenated for sentiment classification. Note that, the transformed labeled data $g(\mathbf{X}_s^l)$ fed to the NP-net are used for sentiment classification and +(positive)/-(negative) pivot predictions simultaneously, but all transformed unlabeled data $g(\mathbf{X}_s^u)$ and $g(\mathbf{X}_t)$ fed to the NP-net can only be used for the +(positive)/-(negative) pivot predictions.

## Hierarchical Content Attention

In the following, we present how to build the document representation progressively from word vectors hierarchically.

**Word Attention** The contextual words contribute unequally to the semantic meaning of a sentence, especially when we focus on a specific task, e.g., sentiment classification. Therefore, we introduce the word-level attention to weight words of each sentence and output with a weighted sum of all words' information.

Assume that a document $\mathbf{x}$ is made up of $n_c$ sentences $\mathcal{C}=\{c_o\}_{o=1}^{n_c}$. Given a sentence $c_o=\{w_{or}\}_{r=1}^{n_w}$, we first map each word into its embedding vector as $e_{or}=Lw_{or}$ throughout an embedding matrix $L$, where $e_{or}\in\mathbb{R}^{n_e\times 1}$. All con-

textual word embedding vectors $\{\{e_{or}\}_{r=1}^{n_w}\}_{o=1}^{n_c}$ are stacked to the external memory $\mathbf{m} \in \mathbb{R}^{n_e \times n_w \times n_c}$, where free memories are padded with zeros vectors. We take each sentence memory $m_o \in \mathbb{R}^{n_e \times n_w}$ and a word-level query vector $q_w$ as the input of the word attention layer. By feeding each word memory $m_{or}$ into a one-layer neural network, we obtain the hidden representation of the $r$th word of the $o$th sentence as:

$$h_{or} = \tanh\left(W_w m_{or} + b_w\right).$$

The importance weight of this word is therefore measured as the similarity between $h_{or}$ and $q_w$, which is further normalized through a mask softmax function as:

$$\alpha_{or} = \frac{M_w(o,r)\exp\left(h_{or}^T q_w\right)}{\sum_{k=1}^{n_w} M_w(o,k)\exp\left(h_{ok}^T q_w\right)},$$

where $M_w(o,r)$ is a word-level mask function in order to avoid the effect of padding vectors. When the word memory $m_{or}$ is occupied, $M_w(o,r)$ equals 1 and otherwise 0. The sentence vector $v_c^o$ is finally output as a weighted sum of all hidden representations $\{h_{or}\}_{r=1}^{n_w}$:

$$v_c^o = \sum_{r=1}^{n_w} \alpha_{or} h_{or}.$$

Note that the word-level query vector $q_w$ is expected to be a high level representation of the query "what is the important word over the sentence for the task". $q_w$ is randomly initialized and jointly learned during the training process.

**Sentence Attention** Similarly, contextual sentences do not contribute equally to the semantic meaning of a document. In light of this, we introduce the sentence-level attention to weight sentences of each document and output with a weighted sum of all sentences' embedding vectors. We calculate the document vector $v_d$ in the same manner upon the sentence vectors $\{v_c^o\}_{o=1}^{n_c}$ we have obtained. We use a sentence-level query vector $q_c$ to acquire the importance weights of sentences. This yields,

$$\begin{aligned} h_o &= \tanh\left(W_c v_c^o + b_c\right), \\ \beta_o &= \frac{M_c(o)\exp\left(h_o^T q_c\right)}{\sum_{a=1}^{n_c} M_c(a)\exp\left(h_a^T q_c\right)}, \\ v_d &= \sum_{o=1}^{n_c} \beta_o h_o, \end{aligned}$$

where $M_c(o)$ is a sentence-level mask function for avoiding the effect of padding sentences. When the sentence memory $m_o$ is completely free, $M_c(o)$ equals 0 and otherwise 1. The sentence-level query vector $q_c$ is expected to be a high-level representation of the query "what is the important sentence over the document for the task". $q_c$ is randomly initialized and jointly learned during the training process.

## Hierarchical Position Attention

The hierarchical content attention model we have described above, however, involves no recurrence and no convolution. To fully take advantage of the order in each sequence, we add positional encodings to our model. Researchers have explored various types of positional encodings (Sukhbaatar et al. 2015; Tang, Qin, and Liu 2016; Gehring et al. 2017; Vaswani et al. 2017), including fixed and learned ones, for different tasks. Different from previous works, we propose a hierarchical positional encoding scheme consisting of a word positional encoding $p_w$ and a sentence positional encoding $p_c$. Such hierarchical positional encodings stay consistent with the hierarchical content mechanism and consider the order information of both words and sentences. In terms of the word positional encoding, we update each piece of memory by adding a vector, i.e., $m_{or} = e_{or} + p_w^r, \forall r \in [1, n_w]$. As for the sentence positional encoding, we add a vector to each sentence's embedding vector, i.e., $v_c^o = \sum_{r=1}^{n_w} \alpha_{or} h_{or} + p_c^o, \forall o \in [1, n_c]$. $p_w$ and $p_c$ shared by the P-net and NP-net are all randomly initialized and jointly learned during the training process.

## Individual Attention Learning

In this section, we introduce loss functions for training the P-net and NP-net, respectively.

**P-net:** P-net is used to learn the domain-shared feature representations that contribute to sentiment classification. Formally, we parameterize the P-net by $H(\mathbf{x};\theta_P)$ which maps a sample $\mathbf{x}$ to a high-level document representation $v_P$. The loss used to train the P-net is made up of two parts:

$$\mathcal{L}_{p-net} = \mathcal{L}_{sen}(H(\mathbf{X}_s^l;\theta_P)) + \mathcal{L}_{dom}.$$

The sentiment loss $\mathcal{L}_{sen}(H(\mathbf{X}_s^l;\theta_P))$ is to minimize the cross-entropy for the labeled data $\mathbf{X}_s^l$ in the source domain:

$$\mathcal{L}_{sen}(H(\mathbf{X}_s^l;\theta_P)) = -\frac{1}{N_s^l}\sum_{i=1}^{N_s^l} y_i \ln \hat{y}_i + (1-y_i)\ln(1-\hat{y}_i),$$

where $y_i \in \{0,1\}$, $\hat{y}_i$ are the groundtruth and sentiment prediction for the $i$th source labeled sample $\mathbf{x}$, respectively.

The domain adversarial loss $\mathcal{L}_{dom}$ enforces the P-net to produce such domain-shared representations that the domain classifier cannot discriminate between domains via the Gradient Reversal Layer (GRL) (Ganin et al. 2016). Mathematically, we define the GRL as $Q_\lambda(x) = x$ with a reversal gradient $\frac{\partial Q_\lambda(x)}{\partial x} = -\lambda I$. As such, the domain classifier can be denoted as $f\left(Q_\lambda(H\left(\mathbf{x};\theta_P\right));\theta_D\right)$ parameterized by $\theta_D$. Learning with a GRL is adversarial: on one hand, the reversal gradient enforces $f$ to be maximized w.r.t. $\theta_P$ for all the data from both domains; on the other hand, $\theta_D$ is optimized by minimizing the cross-entropy domain classification loss:

$$\mathcal{L}_{dom} = -\frac{1}{N_s+N_t}\sum_{i=1}^{N_s+N_t} d_i \ln \hat{d}_i + (1-d_i)\ln\left(1-\hat{d}_i\right),$$

where $d_i \in \{0,1\}$, $\hat{d}_i$ are the groundtruth and the domain prediction for the $i$th sample.

**NP-net:** NP-net is used to discover the domain-specific feature representations for both domains and project them into the domain-shared feature space. We parameterize the NP-net as $H(g(\mathbf{x});\theta_{NP})$ where $\theta_{NP}$ maps a transformed sample $g(\mathbf{x})$ to a high-level document representation $v_{NP}$. The loss function to train the NP-net consists of three terms:

$$\mathcal{L}_{np-net} = \mathcal{L}_{sen}(H(g(\mathbf{X}_s^l);\theta_{NP})) + \mathcal{L}_{pos} + \mathcal{L}_{neg}.$$

The sentiment loss $\mathcal{L}_{sen}(H(g(\mathbf{X}_s^l);\theta_{NP}))$ is formulated to minimize the cross-entropy loss for all the transformed labeled data $g(\mathbf{X}_s^l)$ in the source domain:

$$\mathcal{L}_{sen}(H(g(\mathbf{X}_s^l);\theta_{NP}))=-\frac{1}{N_s^l}\sum_{i=1}^{N_s^l} y_i' \ln \hat{y}_i' + (1-y_i') \ln (1-\hat{y}_i'),$$

where $y_i' \in \{0,1\}$, $\hat{y}_i'$ are the groundtruth and prediction for the $i$th transformed labeled sample $g(\mathbf{x})$ in the source domain, respectively. Moreover, $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ denote the loss functions to minimize the cross-entropy of positive and negative pivot predictions, respectively:

$$\mathcal{L}_{pos}=-\frac{1}{N_s+N_t}\sum_{i=1}^{N_s+N_t} {z_i}^+ \ln \hat{z}_i^+ + (1-{z_i}^+) \ln (1-\hat{z}_i^+),$$

$$\mathcal{L}_{neg}=-\frac{1}{N_s+N_t}\sum_{i=1}^{N_s+N_t} {z_i}^- \ln \hat{z}_i^- + (1-{z_i}^-) \ln (1-\hat{z}_i^-),$$

where ${z_i}^+ \in \{0,1\}$, $\hat{z}_i^+$ are the groundtruth and positive pivot prediction for the $i$th transformed sample, respectively, and ${z_i}^- \in \{0,1\}$, $\hat{z}_i^-$ are the groundtruth and negative pivot prediction for the $i$th transformed sample, respectively.

### Joint Attention Learning

Since the representations of P-net and NP-net are complementary, we conduct joint attention learning for them. For the source labeled data $\mathbf{X}_s^l$ and its transformed data $g(\mathbf{X}_s^l)$, the representation $H(\mathbf{X}_s^l;\theta_P)$ produced by the P-net and the representation $H(g(\mathbf{X}_s^l);\theta_{NP})$ produced by the NP-net are concatenated together for sentiment classification. We combine the losses for both the P-net and NP-net together with a regularizer to constitute the overall objective function:

$$\begin{aligned}\mathcal{L} \ &= \mathcal{L}_{sen}(H(\mathbf{X}_s^l;\theta_P) \oplus H(g(\mathbf{X}_s^l);\theta_{NP})) \\ &\quad + \mathcal{L}_{dom} + \mathcal{L}_{pos} + \mathcal{L}_{neg} + \rho\mathcal{L}_{reg},\end{aligned}$$

where $\rho$ is a regularization parameter to balance the regularization term and other terms. The regularization term $\mathcal{L}_{reg}$ is responsible of avoiding the overfitting by placing the squared $\ell_2$ regularization on parameters for the sentiment classifier, domain classifier, and +(positive)/-(negative) pivot predictors. The goal of the joint attention learning is to minimize $\mathcal{L}$ with respect to all the model parameters except the adversarial training part which will be maximized. All the parameters are optimized jointly with the standard backpropagation algorithm.

## Experiments

In this section, we empirically evaluate the performance of the proposed HATN model.

### Experimental Settings

We conduct the experiments on the Amazon reviews dataset (Blitzer, Dredze, and Pereira 2007), which has been widely used for cross-domain sentiment classification. This dataset contains reviews from five products/domains: Books (B), DVD (D), Electronics (E), Kitchen (K) and Video (V). There are 6000 labeled reviews for each domain with 3000 positive reviews (higher than 3 stars) and 3000 negative reviews (lower than 3 stars), as well as 9750 unlabeled reviews for B, 11843 for D, 17009 for E, 13856 for K and 30180 for V. Note that unlabeled data, which are imbalanced, consist of more positive but less negative reviews. Table 1 summarizes the statistics of the dataset. By following (Pan et al. 2010), we construct 20 cross-domain sentiment classification tasks like A→B, where A corresponds to the source domain and B denotes the target domain. For each pair A→B, we randomly choose 2800 positive and 2800 negative reviews from the source domain A as the training data, the rest from the source domain A as the validation data, and all labeled reviews (6000) from the target domain B for testing.

Table 1: Statistics of the Amazon reviews dataset including the number of training, testing, and unlabeled reviews for each domain as well as the portion of negative samples in the unlabeled data.

| Domain | #Train | #Test | #Unlab. | % Neg. |
|---|---|---|---|---|
| Books | 5600 | 400 | 9750 | 12.70% |
| DVD | 5600 | 400 | 11843 | 12.31% |
| Electronics | 5600 | 400 | 17009 | 12.04% |
| Kitchen | 5600 | 400 | 13856 | 8.08% |
| Video | 5600 | 400 | 30180 | 8.74% |

### Implementation Details

For each transfer pair A→B, we split documents into sentences and tokenize each sentence by NLTK (Bird, Klein, and Loper 2009). The memory size $n_c$ and $n_w$ are set to 20 and 25 respectively. We use the public 300-dimensional *word2vec* vectors with the skip-gram model (Mikolov et al. 2013) to initialize the embedding matrix $L$. They are shared by P-net and NP-net and fine-tuned during the training process. The hidden dimensions of the word attention layer and sentence attention layer are 300. The weights in networks are randomly initialized from a uniform distribution $U[-0.01, 0.01]$. The regularization weight $\rho$ is set to 0.005. For the pivots learned by P-net, we extract only adjectives, adverbs, and verbs with a frequency of at least 5 and remove stop words and negation words.

For training, the model is optimized with the stochastic gradient descent over shuffled mini-batches with momentum rate 0.9. Due to different training sizes for different classifiers, we use a batch size $b_s=50$ for the sentiment classifier, a batch size $b_d=100$ for the domain classifier with a half coming from the source and target domains, respectively, a batch size $b_s$ of source labeled data, and a batch size $b_d$ of unlabeled data from both domains in turn for the +(positive)/-(negative) pivot predictors. Gradients with the $\ell_2$ norm larger than 40 are normalized to be 40. We define the training progress as $p=\frac{t}{T}$, where $t$ and $T$ are current epoch and the maximum one, respectively, and $T$ is set to 100. The learning rate is decayed as $\eta=\max(\frac{0.005}{(1+10p)^{0.75}}, 0.002)$ and the adaptation rate is increased as $\lambda=\min(\frac{2}{1+\exp(-10p)}-1, 0.1)$ during training. We perform early stopping on the validation set during the training process.

Table 2: Classification accuracy on the Amazon reviews dataset

| S | T | Source-only | SFA | DANN | DAmSDA | CNN-aux | AMN | P-Net | NP-Net | HATN | HATN$^h$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| B | D | 0.8057 | 0.8285 | 0.8342 | 0.8612 | 0.8442 | 0.8562 | 0.8685 | 0.8098 | 0.8687 | **0.8707** |
| B | E | 0.7163 | 0.7638 | 0.7627 | 0.7902 | 0.8063 | 0.8055 | 0.8453 | 0.7833 | 0.8545 | **0.8575** |
| B | K | 0.7365 | 0.7810 | 0.7790 | 0.8105 | 0.8338 | 0.8188 | 0.8567 | 0.8042 | 0.8668 | **0.8703** |
| B | V | 0.8145 | 0.8295 | 0.8323 | 0.8498 | 0.8443 | 0.8725 | 0.8648 | 0.8127 | 0.8758 | **0.8780** |
| D | B | 0.7645 | 0.8020 | 0.8077 | 0.8517 | 0.8307 | 0.8453 | 0.8665 | 0.8215 | 0.8720 | **0.8778** |
| D | E | 0.7312 | 0.7600 | 0.7635 | 0.7617 | 0.8035 | 0.8042 | 0.8480 | 0.7865 | 0.8607 | **0.8632** |
| D | K | 0.7343 | 0.7750 | 0.7815 | 0.8260 | 0.8168 | 0.8167 | 0.8570 | 0.8102 | 0.8700 | **0.8747** |
| D | V | 0.8275 | 0.8262 | 0.8595 | 0.8380 | 0.8587 | 0.8740 | 0.8857 | 0.8222 | 0.8845 | **0.8912** |
| E | B | 0.6887 | 0.7235 | 0.7353 | 0.7992 | 0.7738 | 0.7752 | 0.8295 | 0.7608 | 0.8348 | **0.8403** |
| E | D | 0.7260 | 0.7593 | 0.7627 | 0.8263 | 0.7907 | 0.8053 | 0.8202 | 0.7843 | 0.8425 | **0.8432** |
| E | K | 0.8463 | 0.8650 | 0.8453 | 0.8580 | 0.8715 | 0.8783 | 0.8965 | 0.8307 | 0.8955 | **0.9008** |
| E | V | 0.7248 | 0.7565 | 0.7720 | 0.8170 | 0.7878 | 0.8212 | 0.8412 | 0.7648 | **0.8455** | 0.8418 |
| K | B | 0.7153 | 0.7397 | 0.7417 | 0.8055 | 0.7847 | 0.7905 | 0.8353 | 0.7727 | 0.8398 | **0.8488** |
| K | D | 0.7332 | 0.7567 | 0.7532 | 0.8218 | 0.7907 | 0.7950 | 0.8292 | 0.7773 | 0.8437 | **0.8472** |
| K | E | 0.8315 | 0.8538 | 0.8553 | 0.8800 | 0.8673 | 0.8668 | 0.8752 | 0.8342 | 0.8900 | **0.8933** |
| K | V | 0.7608 | 0.7797 | 0.7637 | 0.8147 | 0.7882 | 0.8215 | **0.8542** | 0.7575 | 0.8432 | 0.8485 |
| V | B | 0.7703 | 0.7948 | 0.8003 | 0.8300 | 0.8148 | 0.8350 | 0.8652 | 0.8062 | 0.8703 | **0.8710** |
| V | D | 0.8243 | 0.8365 | 0.8415 | 0.8590 | 0.8525 | 0.8688 | 0.8723 | 0.8147 | 0.8773 | **0.8790** |
| V | E | 0.7187 | 0.7593 | 0.7572 | 0.7767 | 0.8232 | 0.7968 | 0.8432 | 0.7562 | 0.8463 | **0.8598** |
| V | K | 0.7133 | 0.7478 | 0.7522 | 0.7952 | 0.8128 | 0.8098 | 0.8565 | 0.7880 | 0.8577 | **0.8645** |
| Average | | 0.7592 | 0.7869 | 0.7900 | 0.8236 | 0.8198 | 0.8279 | 0.8556 | 0.7949 | 0.8620 | **0.8661** |

## Performance Comparison

The baseline methods in the comparison include:

- **Source-only**: it is a non-adaptive baseline method based on neural networks and uses the most frequent 5000 unigrams and bigrams between domains as features.
- **SFA** (Pan et al. 2010): it is a linear method, which aims to align non-pivots and pivots by spectral feature alignment.
- **DANN** (Ganin et al. 2016): it is based on the adversarial training. DANN performs domain adaptation on the representation encoded in a 5000-dimension feature vector of the most frequent unigrams and bigrams between domains.
- **DAmSDA** (Ganin et al. 2016): it applies DANN on the feature representation generated by the mSDA (Chen et al. 2012). The new representation is the concatenation of the output of the 5 layers and the original input. Each example is encoded as a vector of 30000 dimensions.
- **CNN-aux** (Yu and Jiang 2016): it is based on the CNN (Kim 2014) and makes use of two auxiliary tasks to help induce sentence embeddings.
- **AMN** (Li et al. 2017): it learns domain-shared representations based on memory networks and adversarial training.
- **P-net**: it is the first component of the proposed HATN model without any positional embedding and makes use of the domain-shared representations.
- **NP-net**: it is the second component of the proposed HATN model without any positional embedding and makes use of the domain-specific representations.
- **HATN** & **HATN$^h$**: they are the proposed models that do not contain the hierarchical positional encoding and contain the hierarchical positional encoding, respectively.

Table 2 reports the classification accuracies of different methods on the Amazon reviews dataset. We evaluate our method over 20 transfer pairs, totally 120,000 testing reviews. The proposed HATN$^h$ model consistently achieves the best performance on almost all the tasks. Source-only performs poorly with 75.92% on average due to no adaptive methods applied. SFA only achieves 78.69% on average due to its poor discrete features and a linear classifier used. Besides, SFA highly depends on manual pivot selection methods which may not capture pivots accurately. On the contrary, HATN$^h$ can automatically learn to capture pivots using the P-net attentions. Compared to the adversarial training based approaches, HATN$^h$ outperforms DANN by 7.61%, DAmSDA by 4.25% and AMN by 3.82% on average, respectively. Besides, HATN$^h$ exceeds CNN-aux which still needs to manually select positive and negative pivots by 4.65%. Possible reasons are that HATN can automatically exploit better domain-shared representations with hierarchical attentions and make use of both pivot and non-pivot features which contribute more to the domain-shared representations than those only using the pivot features.

In order to validate the effectiveness of each component, we compare with variants of the proposed HATN$^h$. First, we can see that P-net even outperforms the AMN that considers only word attention by 2.77% on average. which proves that hierarchical attention is more suitable for learning domain-shared representations. To further show the effectiveness of hierarchical attention, we also compare P-net with its variant that considers only sentence attention. P-net with hierarchical attention surpasses P-net with only sentence attention (83.66%) by 1.9% on average. Second, it is reasonable that NP-net only achieves 79.49% on average since the input for NP-net removes all pivots that contribute more to domain-shared features and it is insufficient to do sentiment classification. Third, HATN can get 86.20% on average better than

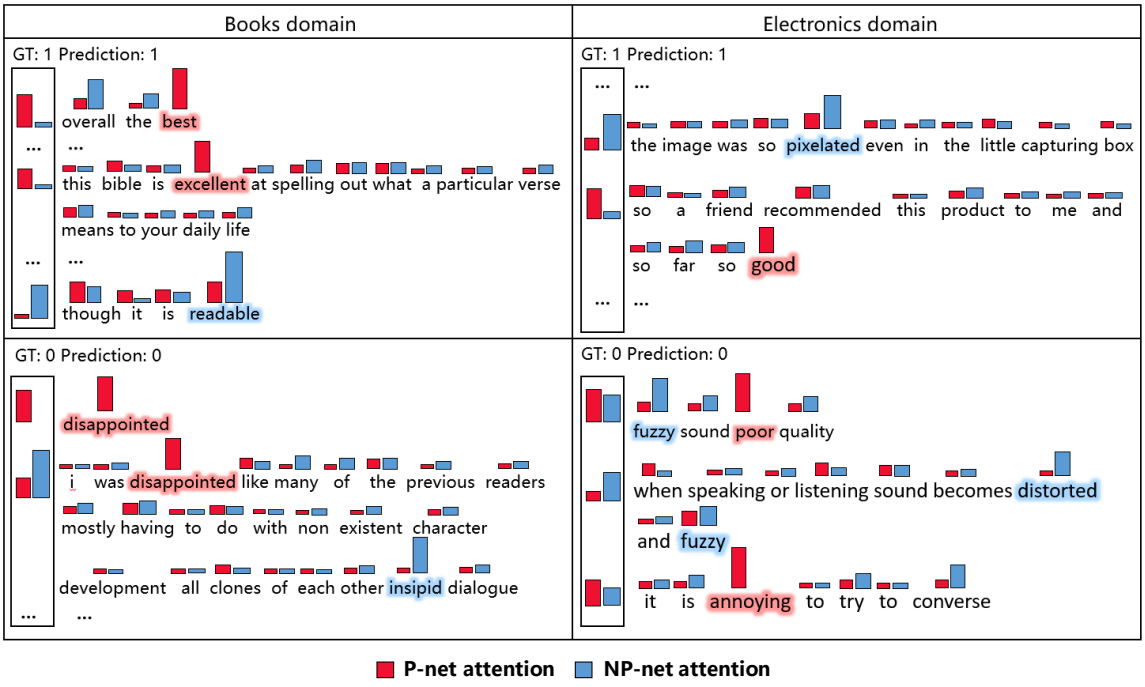


Figure 2: Visualization of attention of the HATN$^h$ in the B→E task. Label 1 denotes positive sentiment and label 0 denotes negative sentiment. NP-net assigns zero attention weights to the pivots due to hiding them.

| | | Books domain | Electronics domain |
|---|---|---|---|
| | | + | - |
| Pivots | | great good excellent best highly wonderful enjoyable love funny fantastic classic favorite interesting loved beautiful amazing fabulous fascinating important nice inspiring well essential useful fun incredible hilarious enjoyed solid inspirational true perfect compelling pretty greatest valuable real humorous finest outstanding refreshing awesome | bad disappointing boring disappointed poorly worst horrible terrible awful annoying misleading confusing useless outdated waste poor flawed simplistic tedious repetitive pathetic hard silly wrong slow weak wasted frustrating inaccurate dull mediocre sloppy uninteresting lacking ridiculous missing difficult uninspired shallow superficial |
| Non-pivots | + | readable heroic believable appealing adorable thoughtful endearing factual inherently rhetoric engaging relatable religious deliberate platonic cohesive genuinely memorable astoundingly introspective conscious grittier inventive entrancing conversational hearted lighthearted eloquent comedic understandable emotional | stereo noticeably noticeable softened rubbery shielded labeled responsive flashy pixelated buffering illuminated personalizing craving glossy matched conspicuous coaxed useable boomy programibilty prerecorded ample fabulously audible intact slick crispier polished markedly intuitive brighter fixable repairable |
| | - | depressing insulting trite unappealing pointless distracting cliched pretentious ignorant cutesy disorganized obnoxious devoid gullible excessively plotless convoluted insipid repetitious formulaic immature trivial sophomoric forgettable hackneyed preachy aimless extraneous implausible monotonous | plugged bulky spotty oily scratched laggy laborious negligible kludgy clogged riled intrusive inconspicuous loosened untoward cumbersome blurry restrictive noisy ghosting corrupted flimsy inferior sticky garbled chintzy distorted patched smearing unfixable ineffective shaky distractingly frayed |

Figure 3: Samples of pivots and non-pivots captured by the HATN$^h$ in the B→E task.

both P-net and NP-net, which proves that the representations of P-net and NP-net are complementary. Moreover, HATN$^h$ can further improve the performance of HATN by 0.41% on average, which also validates that P-net and NP-net can behave better with hierarchical positional encoding.

## Visualization of Attention

In order to validate that our model is able to identity pivots and non-pivots simultaneously with hierarchical attentions, we visualize the word and sentence attention layers of the P-net and NP-net in Figure 2. Figure 2 shows that P-net tends to pay higher word attentions to the pivots between domains, such as positive pivots *best*, *excellent*, *good* and negative pivots *disappointed*, *poor*, *annoying*. The sentences that contain these pivots also get higher sentence attentions in the P-net. Different from P-net, NP-net aims to pay higher word attentions to the non-pivots in the two domains, such as source non-pivots *readable*, *insipid* in the Books domain and target non-pivots *pixelated*, *fuzzy*, *distorted* in the Electronics domain. The sentences that contain these non-pivots also get higher sentence attentions in the NP-net. Overall, the visualization of attentions illustrates that our model can achieve transferring attentions between domains. As showed in Figure 3, we list some examples of pivots and non-pivots captured based on the attention weights of P-net and NP-net respectively in the B→E task. These pivots and non-pivots are crucial for cross-domain sentiment classification.

## Conclusion

In this paper, we propose the HATN method for cross-domain sentiment classification. The proposed HATN can transfer attentions for emotions in both word and sentence levels across domains by automatically capturing pivots and non-pivots, which provides a better interpretability of what to transfer for emotions. Experiments on the Amazon review dataset show the effectiveness of HATN. The proposed hierarchical attention transfer mechanism could be adapted to other domain adaptation tasks such as text classification (Li, Jin, and Long 2012) and machine comprehension (Golub et al. 2017), which are the focus of our future studies.

## Acknowledgement

We thank the support of WeChat-HKUST Joint Lab on Artificial Intelligence Technology, National Grant Fundamental Research (973 Program) of China under Project 2014CB340304 and Hong Kong CERG projects (16211214, 16209715 and 16244616), NSFC (61473087 and 61673202), and the Natural Science Foundation of Jiangsu Province (BK20141340).

## References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 120–128.

Chen, M.; Xu, Z.; Sha, F.; and Weinberger, K. Q. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning*, 767–774.

Ganin, Y., and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, 1180–1189.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(59):1–35.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, 513–520.

Golub, D.; Huang, P.-S.; He, X.; and Deng, L. 2017. Two-stage synthesis networks for transfer learning in machine comprehension.

Haoran Huang, Qi Zhang, X. H. 2017. Mention recommendation for twitter with end-to-end memory network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 1872–1878.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168–177. ACM.

Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Li, Z.; Zhang, Y.; Wei, Y.; Wu, Y.; and Qiang, Y. 2017. End-to-end adversarial memory network for cross-domain sentiment classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2237–2243.

Li, L.; Jin, X.; and Long, M. 2012. Topic correlation analysis for cross-domain text classification. In *AAAI*.

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5(1):1–167.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, 3111–3119.

Pan, S. J.; Ni, X.; Sun, J.-T.; Yang, Q.; and Chen, Z. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, 751–760. ACM.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 79–86. Association for Computational Linguistics.

Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, volume 1631, 1642.

Sukhbaatar, S.; Szlam, A.; Weston, J.; and Fergus, R. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, 2440–2448.

Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2015. Target-dependent sentiment classification with long short term memory. *CoRR, abs/1512.01100*.

Tang, D.; Qin, B.; and Liu, T. 2016. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *CoRR* abs/1706.03762.

Wang, S., and Manning, C. D. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 90–94.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1480–1489.

Yu, J., and Jiang, J. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.