# End-to-End Adversarial Memory Network for Cross-domain Sentiment Classification

**Zheng Li, Yu Zhang[†], Ying Wei[†], Yuxiang Wu, Qiang Yang**
Department of Computer Science and Engineering, Hong Kong University of Science and Technology
{zlict,zhangyu,yweiad,ywubw,qyang}@cse.ust.hk [*]

## Abstract

Domain adaptation tasks such as cross-domain sentiment classification have raised much attention in recent years. Due to the domain discrepancy, a sentiment classifier trained in a source domain may not work well when directly applied to a target domain. Traditional methods need to manually select *pivots*, which behave in the same way for discriminative learning in both domains. Recently, deep learning methods have been proposed to learn a representation shared by domains. However, they lack the interpretability to directly identify the pivots. To address the problem, we introduce an end-to-end Adversarial Memory Network (AMN) for cross-domain sentiment classification. Unlike existing methods, the proposed AMN can automatically capture the pivots using an attention mechanism. Our framework consists of two parameter-shared memory networks with one for sentiment classification and the other for domain classification. The two networks are jointly trained so that the selected features minimize the sentiment classification error and at the same time make the domain classifier indiscriminative between the representations from the source or target domains. Moreover, unlike deep learning methods that cannot tell which words are the pivots, AMN can offer a direct visualization of them. Experiments on the Amazon review dataset demonstrate that AMN can significantly outperform state-of-the-art methods.

## 1 Introduction

Sentiment classification is an important task in natural language processing and is essential to understand user opinions in social networks or product reviews [Pang and Lee, 2008; Liu, 2012]. This task aims to identify the overall sentiment polarity (e.g., positive or negative) of a document. Traditional approaches for sentiment classification are based on support vector machine with handcrafted features such as bag-of-n-grams [Wang and Manning, 2012]. Recently, deep learning models [Socher *et al.*, 2013; Tang *et al.*, 2015] are exploited in sentiment classification to automatically learn a good representation. Unfortunately, effective deep learning methods are highly dependent on large amounts of labeled training data which requires time-consuming and expensive manual labeling. In order to alleviate the dependence on a large number of labeled data, cross-domain sentiment classification, which utilizes labeled data from related domains, becomes a promising direction.

Over the last decade, many methods have been proposed for cross-domain sentiment classification. Blitzer *et al.* [Blitzer *et al.*, 2007] proposed a Structural Correspondence Learning (SCL) method to learn a joint low-dimensional feature representation for the source and target domains. Similarly, Pan *et al.* [Pan *et al.*, 2010] proposed a Spectral Feature Alignment (SFA) method to align the pivots with the non-pivots to build a bridge between the source and target domains. However, these methods need to tediously select the pivots based on criterions such as the frequency in both domains [Blitzer *et al.*, 2006], the mutual information between features and labels on the source domain data [Blitzer *et al.*, 2007], and the mutual information between features and domains [Pan *et al.*, 2010]. Different from these methods, Glorot *et al.* [Glorot *et al.*, 2011] proposed a Stacked Denoising Autoencoders (SDA) to automatically learn a unified feature representation for documents from a large amount of data in all the domains. Similar to this work, Chen *et al.* [Chen *et al.*, 2012] proposed a Marginalized Stacked Denoising Autoencoders (mSDA) model to improve SDA in terms of the speed and scalability to high-dimensional data. Ganin *et al.* [Ganin and Lempitsky, 2015; Ganin *et al.*, 2016] proposed the Domain-Adversarial training of Neural Networks (DANN) for domain adaptation to achieve promising results on benchmark datasets. They use a gradient reversal layer to reverse the gradient direction in order to produce representations such that a domain classifier cannot predict the domain of the encoded representation, and at the same time, a sentiment classifier is built on the representation shared by domains to reduce the domain discrepancy and achieves better performance for cross-domain sentiment classification. However, these deep models lack interpretability to directly identify the pivots.

In order to improve the interpretability of deep models, we propose an end-to-end Adversarial Memory Network (AMN) for cross-domain sentiment classification. Our approach can

[*][†]These two authors contributed equally to this work.

automatically capture the pivots using an attention mechanism without the manual selection. Actually, the memory network can capture the corresponding important words according to the task of interest using the attention mechanism. In order to capture these pivots automatically, we make use of their characteristics that pivots are the important sentiment words for sentiment classification and are shared in both domains. Specifically, our framework consists of two parameter-shared memory networks, where one network is for sentiment classification and the other is for domain classification. The two networks are jointly trained so that the selected features can minimize the sentiment classification error and in the meanwhile, make the representations from the source and target domains indiscriminative for the domain classifier. In this way, the proposed AMN can focus on learning pivots. Experiments on the Amazon reviews benchmark dataset demonstrate that AMN outperforms state-of-the-art methods.

Our contributions are summarized as follows:

- The proposed AMN model can automatically capture the pivots using the attention mechanism without manually selecting pivots.
- Unlike deep learning based methods that cannot tell us which words are the pivots, the proposed AMN model can offer a direct visualization of them, which makes the representations shared by domains more interpretable.
- Empirically the proposed AMN method can achieve better performance than the state-of-the-art methods.

## 2 Related Works

Traditional methods need to manually select pivots. For example, the SCL method [Blitzer *et al.*, 2007] is proposed to learn a low-dimensional feature representation for source and target domains. The efficacy of SCL depends on the choice of pivots and it assumes that pivots are frequently occurring words in both domains and they are also good predictors of source domain labels. Thus, they select pivots with highest mutual information between features and source labels. Similarly, the SFA method [Pan *et al.*, 2010] aims to build a bridge between source and target domains by aligning pivots to non-pivots. They argue that the pivot selection method of SCL can help identify features relevant to the source labels but there is no guarantee that the selected features act similarly in both domains.

Recently, some efforts have been initiated based on deep learning models for cross-domain sentiment classification. The SDA model [Glorot *et al.*, 2011] is the first to automatically learn a unified feature representation for documents from large amounts of data in all the domains. The mSDA model [Chen *et al.*, 2012] addresses the high computational cost and scalability problem of SDA. However, these two methods just make use of the SDA model to exploit a unified feature representation for all the domains without identifying the pivots. Similarly, the DANN model [Ganin and Lempitsky, 2015; Ganin *et al.*, 2016], which uses a gradient reversal layer to produce representations, cannot identify the pivots.

## 3 Adversarial Memory Network

In this section, we introduce the proposed AMN model for cross-domain sentiment classification. We first give the problem definition and notations. Then we give an overview of the AMN model. Finally we present the details of different components as well as the training process.

### 3.1 Problem Definition and Notations

Suppose we have a set of labeled data $X_s^l = \left\{x_s^i, y_s^i\right\}_{i=1}^{N_s^l}$ as well as some unlabeled data $X_s^u = \left\{x_s^i\right\}_{i=N_s^l+1}^{N_s^l+N_s^u}$ in a source domain, where $N_s^l$ and $N_s^u$ denote the number of labeled data and unlabeled data, respectively. In a target domain, there is a set of unlabeled data $X_t = \left\{x_t^i\right\}_{i=1}^{N_t}$ , where $N_t$ is the number of unlabeled data. The task of the cross-domain sentiment classification is to learn a robust classifier trained on labeled data in the source domain to predict the polarity of unlabeled examples from the target domain.

### 3.2 An Overview of the AMN Model

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

The AMN model is inspired by the successful use of memory network in question answering [Sukhbaatar *et al.*, 2015], document classification [Yang *et al.*, 2016], and aspect-level sentiment classification [Tang *et al.*, 2016]. The memory network can capture the corresponding important words according to the task of interest using the attention mechanism. The goal of the AMN model is to automatically capture the pivots. Therefore, we need to make use of the memory network to extract these pivots that have two attributes: (1) They are the important sentiment words for sentiment classification; (2) These words are shared in both domains. In order to achieve this goal, we design two parameter-shared deep memory networks, where one network, denoted by MN-sentiment, is for sentiment classification and the other, denoted by MN-domain, is for domain classification that aims to predict domain labels of samples, i.e., coming from the source or target domain. These two networks are jointly trained so that the selected features minimize the sentiment classification errors and at the same time make a domain classifier incapable of discriminating samples from the source and target domains. The overall architecture of the AMN model is shown in Figure 1.

Given a document $d = \{w_1, w_2 \dots w_n\}$, we first map each word into its embedding vector as $e_i = Aw_i$ and get a vector representation $e = \{e_1, e_2 \dots e_n\}$ for the document. These word vectors are stacked and put into the external memory $\mathbf{m} \in \mathbb{R}^{\mathbf{d}\times\mathbf{m}}$, where $m$ is the memory size that is larger than the maximum length of documents and the free memories are padded with zero vectors. As illustrated in Figure 1, each memory network contains multiple hops, each of which consists of an attention layer and a linear layer. In the first hop, we use a query vector $q_w$ as the input to capture important words from memory $\mathbf{m}$ through the attention layer. The query vector $q_w$ is randomly initialized during the training process and can be learned for a high-level representation according to the task of interest. The output of the attention layer and

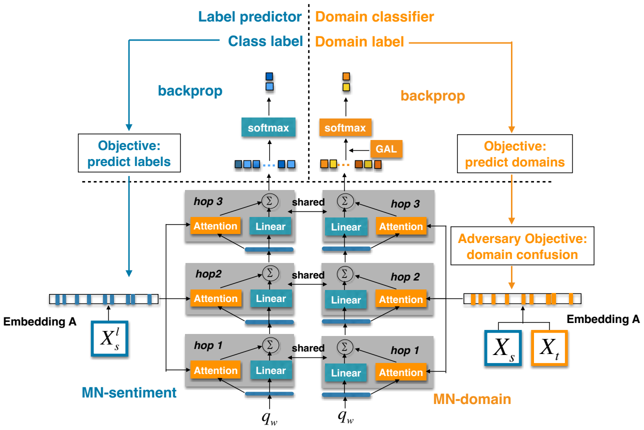

Figure 1: The framework of the AMN model.

the linear transformation of the query vector are combined as the input of the next hop. The output vectors in the last hop of memory networks are considered as representations of a document with respect to the queries, and they are further used as the representations for the sentiment classifier and domain classifier respectively.

For the MN-sentiment network, the query vector $q_w$ can be seen as a high-level representation of a fixed query 'what is the sentiment word' over words. For the MN-domain network, we add the Gradient Reversal Layer (GRL) [Ganin and Lempitsky, 2015; Ganin *et al.*, 2016] between the last hop of the MN-domain network and the domain classifier to reverse the gradient direction of the MN-domain network. In this way, the MN-domain network can produce a representation such that the domain classifier cannot predict the domain of the encoded representation and hence maximize the domain confusion. Thus, the query vector $q_w$ used by the MN-domain network can be seen as a high-level representation of a fixed query 'what is the domain-shared word'. These two memory networks share all the parameters including the query vector $q_w$ and they are jointly trained such that the query vector $q_w$ stands for a higher-level representation of a fixed query 'what is the pivot'. Besides, the parameters of the attention and linear layers are shared in different hops for each memory network.

## 3.3 Components

In the following, we introduce components in AMN one by one.

### Word Attention

Actually, not all words contribute equally to the representation of a document for different tasks, either sentiment classification or domain classification. Therefore, we introduce an attention mechanism to extract such important words for the task and aggregate the representation of those meaningful words to form an output of the attention model.

We take the external memory $\mathbf{m} \in \mathbb{R}^{\mathbf{d}\times\mathbf{m}}$ and the query vector $q_w$ as the input of a memory network. For the MN-sentiment network, we only update the external memory $\mathbf{m_s}$ by the labeled data in the source domain. For the MN-domain network, we update the external memory $\mathbf{m_d}$ by all the data from the source and target domains. In the following, we denote by $\mathbf{m}$ as one of the two external memories.

We first feed the each piece of memory $m_i$ through a one-layer neural network to get a hidden representation $h_i$, and measure the importance of the word as the similarity of $h_i$ with the query vector $q_w$. Then we get a normalized importance weight $a_i$ through a softmax function as

$$a_i = \frac{\exp\left(h_i^T q_w\right)}{\sum_{j=1}^{n} \exp\left(h_j^T q_w\right)},$$

where $n$ is the size of the memory occupied and $h_i = \tanh\left(W_s m_i + b_s\right)$. Here we do not use the whole memory $\mathbf{m}$ because we have found that the attention model sometimes assigns large weights to the free memories and gives low weights to the occupied part, which may reduce the quality of the document representation. The weights $a_i$ for free memories are all set by zero.

After that, we compute an output vector $v$ of the attention layer as a weighted sum of each piece of memory in $\mathbf{m}$:

$$v = \sum_{i=1}^{m} a_i m_i.$$

Then, we get the output vectors $v_s$ and $v_d$ in the last hop of each memory network as the feature representations produced by the MN-sentiment and MN-domain networks, respectively. Note that, we also add the Position Encoding (PE) introduced in [Sukhbaatar *et al.*, 2015] to get the final external memory $\mathbf{m}$.

### Sentiment Classifier

As illustrated in the top left part of Figure 1, we treat the output vector in the last hop of the MN-sentiment network as the document representation $v_s$ and feed it to the softmax layer for sentiment classification:

$$y = \mathrm{softmax}\left(W_s v_s + b_s\right).$$

The goal of the sentiment classifier is to minimize the cross-entropy for all the labeled data in the source domain as

$$L^{sen} = -\frac{1}{N_s^l} \sum_{i=1}^{N_s^l} \left(\hat{y}_i \ln y_i + (1 - \hat{y}_i) \ln\left(1 - y_i\right)\right),$$

where $\hat{y}_i, y_i \in \{0, 1\}$ are the ground truth and the predicted sentiment label for sample $i$, respectively.

### Domain Classifier

Similarly, as showed in the top right part of Figure 1, we treat the output vector in the last hop of the MN-domain network as the document representation $v_d$ for domain classification. Before feeding $v_d$ to the softmax layer, the document representation $v_d$ goes through the GRL. During the forward propagation, the GRL acts as an identity function but during the backpropagation, the GRL takes the gradient from the subsequent level, multiplies it by $-\lambda$ and passes it to the preceding layer. We can formulate GRL as a 'pseudo-function' $Q_\lambda\left(x\right)$

by two equations below in order to describe its forward- and backward- behaviours:

$$Q_\lambda(x) = x$$
$$\frac{\partial Q_\lambda(x)}{\partial x} = -\lambda I.$$

We refer the document representation $v_d$ thorough the GRL as $Q_\lambda(v_d) = \hat{v_d}$ and then feed it to the softmax layer as

$$d = \mathrm{softmax}\left(W_d \hat{v_d} + b_d\right)$$

Mathematically, we treat the MN-domain network as $v_d = H(x; \theta_h)$ and the domain classifier as $Z(Q_\lambda(v_d); \theta_z)$. Learning with the GRL is adversarial such that $\theta_z$ is optimized to increase Z's ability to distinguish between document representations from the source and target domains, while $\theta_h$ learns document representations to reduce the domain classification accuracy due to the reversal of the gradient. Essentially, $\theta_h$ is optimized to maximize the loss of the domain classifier, while simultaneously optimizing the parameters $\theta_z$ of the domain classifier to minimize the loss of the domain classifier. The domain classifier is trained to minimize the cross-entropy for all data from the source and target domains:

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

where $\hat{d_i}, d_i \in \{0, 1\}$ are the ground truth and the predicted domain label for sample $i$, respectively and $N_s = N_s^l + N_s^u$ is the number of data from the source domain.

**Regularization**

In order to avoid the overfitting problem for the sentiment classifier and the domain classifier, we also add the squared Frobenius norm for weights $W_s, W_d$ and squared $\ell_2$ norm regularization for bias terms $b_s, b_d$:

$$L^{reg} = \|W_s\|_F^2 + \|W_d\|_F^2 + \|b_s\|_2^2 + \|b_d\|_2^2$$

where $\|\cdot\|_2$ denotes the $\ell_2$ norm of a vector and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

### 3.4 Joint learning

We combine each component losses into an overall object function:

$$L^{total} = L^{sen} + L^{dom} + \rho L^{reg}$$

where $\rho$ is a regularization parameter to balance the regularization term and other loss terms. The goal of the joint learning is to minimize $L^{total}$ with respect to the model parameters except for the adversarial training part. The sentiment classifier is trained to minimize the sentiment classification loss $L^{sen}$ for the sentiment classification task. Because the unsupervised domain adaptation setting is adopted, the sentiment classification loss is only applied to the source domain. It would be convenient to add the loss term for the labeled data in the target domain if they are available. The domain classification loss $L^{dom}$ makes use of both labeled and unlabeled data from both domains. Each mini-batch for the domain classifier is balanced, half coming from the source and half from the target. The regularization term $L^{reg}$ is added to avoid the overfitting. All the parameters are optimized jointly by using the standard backpropagation algorithm.

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 | 1600 | 400 | 6000 | 13.45% |
| DVD | 1600 | 400 | 34741 | 21.47% |
| Electronics | 1600 | 400 | 13153 | 11.92% |
| Kitchen | 1600 | 400 | 16785 | 17.82% |

## 4 Experiment

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

### 4.1 Settings

Experiments are conducted on the Amazon reviews dataset [Blitzer *et al.*, 2007], 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 appliances (K). There are 2000 labeled reviews for each domain with 1000 positive reviews (higher than 3 stars) and 1000 negative reviews (3 stars or lower), as well as 6000 unlabeled reviews for B, 34741 for D, 13153 for E, and 16785 for K. Note that unlabeled data is imbalanced consisting of more positive but less negative reviews. Table 1 summarizes the statistics of the dataset.

By following [Pan *et al.*, 2010], we construct 12 cross-domain sentiment classification tasks: D→B, E→B, K→B, K→E, D→E, B→E, B→D, K→D, E→D, B→K, D→K, E→K, where the word before the arrow corresponds to the source domain and the word after the arrow corresponds to the target domain. For each transfer pair A→B, we randomly choose 800 positive and 800 negative reviews from the source domain A as the training data, the rest from the source domain A as the validation data, and 200 positive and 200 negative reviews from the target domain B for testing. All data (labeled and unlabeled data) from both domains is used for domain classifier.

### 4.2 Implementation Details

For each transfer pair A→B, the word embedding in A is first initialized with the public 300-dimensional *word2vec* vectors that are trained on 100-billion-word Google News using the continuous bag-of-words architecture [Mikolov *et al.*, 2013] and they are fine-tuned during the training process. The weights in networks are randomly initialized from a uniform distribution $U[-0.01, 0.01]$. The memory size $m$ is set to 500 and the number of hops is 3. The regularization weight $\rho$ is set to 0.05, which is obtained via 5-fold cross-validation on the labeled data in the source domain and is used for all transfer pairs.

The model is optimized with the stochastic gradient descent over shuffled mini-batches with momentum rate 0.9. Due to different training sizes for the sentiment classifier and domain classifier, we set the batch size $b_d$ for the domain classifier with 100, half coming from the source and target domains, and use the same number of batches for both classifiers. 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 120. By following [Ganin *et al.*, 2016], the learning rate is decayed as $\eta = \frac{0.0075}{(1+10p)^{0.75}}$ and the adaptation rate is increased as $\lambda = \frac{2}{1+exp(-10p)} - 1$ during training. We perform early stopping on the validation set during the training process.

### 4.3 Performance Comparison

The baseline methods in the comparison include:

- **SCL**: Blitzer et al. proposed Structural Correspondence Learning (SCL) to learn a low-dimensional feature representation for source and target domains [Blitzer *et al.*, 2007].
- **SFA**: Pan et al. proposed Spectral Feature Alignment (SFA) to build a bridge between source and target domains by aligning pivots with non-pivots [Pan *et al.*, 2010].
- **DANN**: Ganin *et al.* have applied the shallow version of Domain Adversarial Neural Networks (DANN) to the cross-domain sentiment classification [Ganin *et al.*, 2016]. The DANN performs domain adaptation on the review representation encoded in a 5000-dimension feature vector of unigrams and bigrams and is a baseline method for the adversarial training.
- **DAmSDA**: Ganin *et al.* have also applied their shallow version of DANN on the feature representation generated by Marginalized Stacked Denoising Autoencoders (mSDA). 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. Here we denote it as DAmSDA.
- **DACNN**: We also apply the shallow version of DANN on the representation generated by the CNN-non-static version of the Convolutional Neural network (CNN) [Kim, 2014]. Here, we refer it as DACNN.

For the SCL, DANN and DAmSDA methods, we use the source codes provided by original authors and we reimplement other baseline methods.

Figure 2 reports the classification accuracies of different methods on the Amazon reviews dataset. The proposed AMN model consistently achieves the best performance on almost all the tasks. SCL and SFA perform poorly on average. Their performance is highly dependent on pivots selection methods which may not capture the pivots accurately. On the contrary, AMN can automatically capture the pivots with the attention mechanism. Compared to the adversarial training based approaches, AMN outperforms DANN by 7.71% and exceeds DAmSDA and DACNN by 4.38% and 4.36% on average, respectively. One reason is that AMN can automatically capture the pivots and assign them higher weights to generate a better feature representation shared by domains. Besides, DAmSDA has a high computational cost because it depends on high-dimensional features. For AMN, the two memory networks share parameters and the parameters of attention and linear layers are also shared in different hops for each memory network. In addition, the dimension of the features shared by the source and target domain is 300, which is only 1% of that of DAmSDA. Hence, AMN is superior in terms of both classification accuracy and computational efficiency. We also compare with a variant of the proposed AMN model without domain adaptation, denoted by AMN-NA. That is, the AMN-NA model uses labeled data in the source domain for training and then directly tests its performance on the target domain. It is interesting that AMN-NA can even perform comparably against baseline methods. It is because the memory network can capture important words for sentiment classification. As a reference, we also provide a upper-bound of the performance, which corresponds to horizontal lines in Figure 2, by training AMN-NA on the labeled data in the target domain. According to Figure 2, we can see that in all the settings, the performance of AMN is close to the upper-bound and sometimes even better, which demonstrates the effectiveness of AMN.

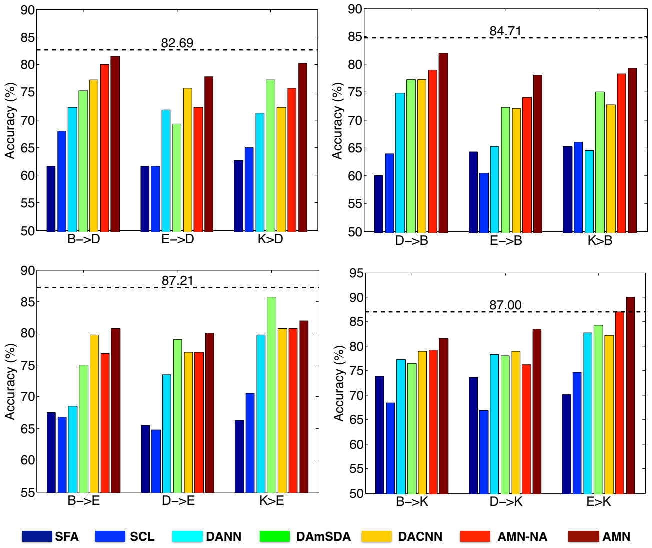

Figure 2: Average results for cross-domain sentiment classification on the Amazon reviews dataset.

### 4.4 Visualization of Attention

In order to validate that our model is able to select pivots in a document, we visualize the final attention layer of the MN-sentiment network in Figure 3. Figure 3 shows that our model can capture important sentiment words shared in both domains, such as positive sentiment words such as *great*, *good*, *fantastic*, *elegant*, *best*, *excellent*, *gorgeous*, *beautiful* and negative sentiment words including *terrible*, *poor*, *uncomfortable*, *disappointed*, *disappointment*, *unusable*, *useless*.

As shown in Figure 4, we list some examples of pivots captured based on the attention weights in the E→K task. These pivots contribute a lot to the representations shared by domains and are crucial to cross-domain sentiment classification.

GT:1 Prediction:1
great dvd media i have burned over 100 of these in the past 6 months i have only had 1 burn badly havent found a dvd player yet that they wont play in

GT:1 Prediction:1
good for canon a95 fantastic take all the videos and pictures you want with the best quality

GT:1 Prediction:1
you cannot beat a belkin cable great quality excellent construction and strong rj45 plugs i have worked with a decent share of cat5 and i have never had to cut and terminate a belkin cable due to regular wear and tear

GT:0 Prediction:0
i cant hear you sound output is terrible you cant hear it in a car or airplane with high quality noise cancelling earphones when i called customer service they told me it was not intended for use in a car or airplane picture is very good but i have heard better sound from much cheaper players dont waste your money

GT:0 Prediction:0
great technology terrible customer experience i had the same exact experience with the poor fit of these headphones and the rude customer service their surround sound he592 phones dont fit well either

GT:0 Prediction:0
uncomfortable i had these headphones for a few years then they got crushed in half in my bag they hurt your ears after about ten minutes they are durable though i would recommend the kind that clip behind your ear

(a) Electronics domain

GT:1 Prediction:1
great gifts i love the rapid ice wine coolers i give them for token gifts and use them frequently myself they are great for a spure of the moment glass of wine that needs chilling

GT:1 Prediction:1
an elegant way of serving its a traditional serve ware for serving the soup course the color of the tureen set allows it to be used with many of the dinnerwares amp the size is adequate to serve at least 810 people the under plate is something not found with usual tureen sets which gives it an elegant look but it appears a little overpriced

GT:1 Prediction:1
gorgeous i just received this as a wedding gift and it is beautiful a great gift

GT:0 Prediction:0
disappointed whisker i am usually very pleased with oxo products but this one is a big disappointment i have not found it to be good for or at anything wished id saved the five bucks

GT:0 Prediction:0
too poorly made for everyday use we have a full line of fiesta dishware and thought having the matching flatware would be nice after a year of standard use and dishwashing about 13 of the flatware is unusable the upside is that it is cheap and replaceable but count me among those who would rather pay more for something that lasts we are in the process of ditching the fiesta flatware line and moving to something more robust

GT:0 Prediction:0
totally useless we bought this to use at events for a chocolate themed group at college and used it several times before giving up

(b) Kitchen domain

Figure 3: Samples from the Amazon reviews dataset in the E→K task. Deeper color implies larger attention weights. Label 1 denotes positive sentiment and label 0 denotes negative sentiment.

| Tasks | Positive sentiment words | Negative sentiment words |
|---|---|---|
| Electronics -kitchen | good great amazing excellent better best nice cool perfect happy fantastic outstanding cheaper easy beautiful convenient well fine wonderful worthwhile pleased affordable fast cheap flawless unbelievable reliable satisfied impressive pretty compatible nicely comfort powerful brilliant worth unbreakable fancy impressed compact handy elegant quick love durable | bad worst worse uncomfortable useless confused unreliable sad unacceptable poor impossible misleading unhappy waste upset disappointing thrilled disappointed disappointment negative terrible messy unsuitable worthless horrible poorly pricy defective dangerous fragile incorrectly stressful confusing expensive frustrating difficult unexpected painful ridiculous |

Figure 4: Samples of pivots captured by AMN in the E→K task.

### 4.5 Visualization of Representation

Figure 5 shows the visualization of the feature representation of the AMN model for the training data in the source domain and the testing data in the target domain for the E→K tasks. As shown in Figure 5, two feature distributions from the source and target domains are very similar, implying that the learned feature representation can be well shared by both domains.

## 5 Conclusion

In this paper, we propose the AMN model for cross-domain sentiment classification. The AMN model can automatically capture the pivots by using an attention mechanism without manually selecting pivots. AMN can offer a direct visualization of the pivots, which increases the interpretability of AMN. These pivots are assigned with higher weights to generate a better feature representation shared by domains. Experiments on the Amazon review dataset demonstrate that AMN can significantly outperform the state-of-the-art methods.

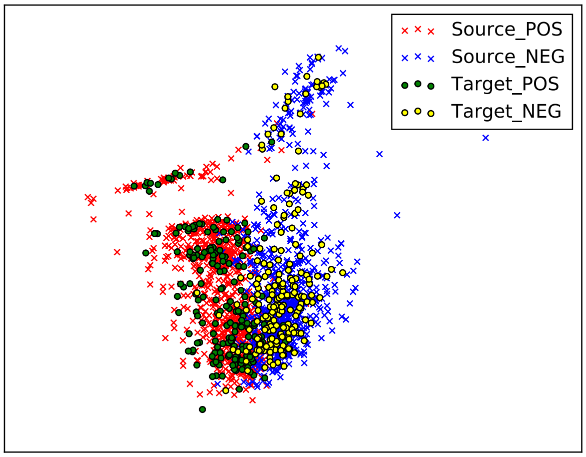

Figure 5: Visualization by applying principal component analysis to the representation of source training data and target testing data produced by AMN for E→K tasks.

Due to the good compatibility of memory network, the proposed AMN model could be easily adapted to other domain adaptation tasks such as POS tagging [Blitzer *et al.*, 2006] and relation extraction [Bollegala *et al.*, 2011], which are the focus of our future studies.

## Acknowledgements

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

## References

[Blitzer *et al.*, 2006] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, 2006.

[Blitzer *et al.*, 2007] 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 for Computational Linguistics*, 2007.

[Bollegala *et al.*, 2011] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relation adaptation: Learning to extract novel relations with minimum supervision. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011.

[Chen *et al.*, 2012] Minmin Chen, Zhixiang Xu, Fei Sha, and Kilian Q Weinberger. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning*, pages 767–774, 2012.

[Ganin and Lempitsky, 2015] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1180–1189, 2015.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520, 2011.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

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

[Mikolov *et al.*, 2013] 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 26*, pages 3111–3119, 2013.

[Pan *et al.*, 2010] 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.

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

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 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, page 1642, 2013.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, pages 2440–2448, 2015.

[Tang *et al.*, 2015] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Target-dependent sentiment classification with long short term memory. *CoRR, abs/1512.01100*, 2015.

[Tang *et al.*, 2016] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*, 2016.

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

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 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*, pages 1480–1489, 2016.