Cross-Domain Sentiment Classification via A Bifurcated-LSTM

Jinlong Ji, Changqing Luo, Xuhui Chen, Lixing Yu, and Pan Li

Case Western Reserve University, Ohio, USA
{jxj405,cxl881,xxc296,lxy257,pxl288}@case.edu

Abstract. Sentiment classification plays a vital role in current online commercial transactions because it is critical to understand users' opinions and feedbacks in businesses or products. Cross-domain sentiment classification can adopt a well-trained classifier from one source domain to other target domains, which reduces the time and efforts of training new classifiers in these domains. Existing cross-domain sentiment classification methods require data or other information in target domains in order to train their models. However, collecting and processing new corpora require very heavy workload. Besides, the data in target domains may be private and not always available for training. To address these issues, motivated by multi-task learning, we design a Bifurcated-LSTM which takes advantages of attention-based LSTM classifiers along with augmented dataset and orthogonal constraints. This Bifurcated-LSTM can extract domain-invariant sentiment features from the source domain to perform sentiment analysis in different target domains. We conduct extensive experiments on seven classic types of product reviews, and results show that our system leads to significant performance improvement.

1 Introduction

Sentiment classification plays a vital role in current online commercial transactions because it is essential to understand users' opinions and feedbacks in businesses or products. It identifies the overall sentiment polarity (e.g., positive or negative) of a text. In 2002, Bo et al. [22] were the first pioneers to utilize machine learning techniques to tackle the sentiment classification problem. Since then, many researchers have shown their interests in this field [9, 21]. Noticeably, most of them try to obtain sentiment classifiers by assuming there are sufficient training data in a specified domain. In practice, consumers are usually interested in a number of different types of product, and sentiment is expressed differently in various domains. When we apply previous sentiment classification techniques, large amounts of labeled data are required each time when we need to conduct sentiment analysis for a new product. To alleviate this issue, cross-domain sentiment classification [4], which utilizes labeled data from related domains, has attracted people's attention. It is to adapt a well-designed sentiment classifier, which is trained on the data in one domain, to classify the sentiment of data in other domains.

Although in the literature, several cross-domain sentiment classification schemes have been proposed [11, 13], all of them need target domain data, which is not always available. Specifically, when a new domain emerges, it costs a lot of efforts to collect and process its data, especially for supervised methods where the labels have to be added manually. Besides, there may be sensitive information in the new domain data, such as reviews for beta version products, which cannot be leaked or made public.

To address these problems, we design a novel Bifurcated-LSTM for cross-domain sentiment classification. Particularly, we notice that there are two crucial points a user's review tries to convey: topic and sentiment. Topic, which is different from one domain to another, describes the product or service that the customer comments on. Sentiment is the opinion of the customer about the topic, which is common in all the domains, such as "positive" or "negative". By eliminating the topic-related features, we can decrease the topic-conglutination influence from the source domain to the target domain. Motivated by the idea of multi-task learning, which can detach each task's private feature space from the shared space among several tasks [16], the proposed Bifurcated-LSTM divides the review representation feature space into topic subspace and sentiment subspace. After that, the extracted domain-invariant sentiment features from the source domain can be utilized to perform sentiment analysis in different target domains. To better capture domain-dependent topic features from the source domain training dataset, we apply the dataset augmentation method to improve the performance. Besides, to prevent the topic and sentiment feature spaces interfering with each other, we introduce orthogonal constraints strategies. The experiment results show that our approach can improve sentiment classification in each target domain.

The main contributions of this paper are four-folds:

- We design a novel Bifurcated-LSTM that divides a sentence feature space into domain-dependent topic feature space and domain-independent sentiment feature space.
- We use dataset augmentation to better extract domain-dependent topic features from the source domain, which can help separate these features from sentiment features.
- We employ orthogonal constraint technique to avoid interference between topic and sentiment features.
- Different from other cross-domain sentiment classification models, our system no longer needs any target domain data or other related information.

2 Related Works

2.1 Cross-Domain Sentiment Classification

Cross-domain sentiment classification, a subclass of domain adaptation, is to first learn a sentiment classifier for a source domain by training on this domain's data and then apply the learned classifier into other domains (i.e., target domains) for sentiment classification. To achieve high accuracy, one main challenge is how to analyze data from the source domain and identify its feature space that happens to be related to the feature space of a target domain.

Previous works have studied the problem of feature space mapping from a source domain to target domains [3, 4]. However, those works require the data from target domains and need a lot of efforts to label data manually.

2.2 Multi-task Learning

Multi-task learning is to learn multiple related tasks in parallel so as to improve the learning performance. In particular, the representations of all tasks are effectively combined by neural-based models. The architecture of multi-task learning is shown in Figure 1. Specifically, multiple tasks have several shared layers that are used to detach common feature space. Then, the output of the shared layers is split into multiple branches that are utilized to capture private features for each task [16].

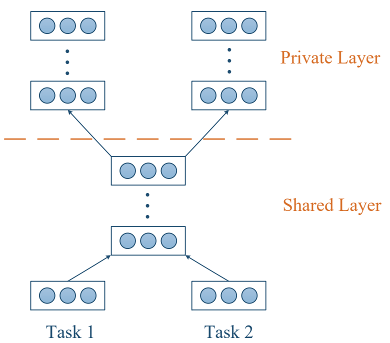

Fig. 1. The architecture of multi-task learning.

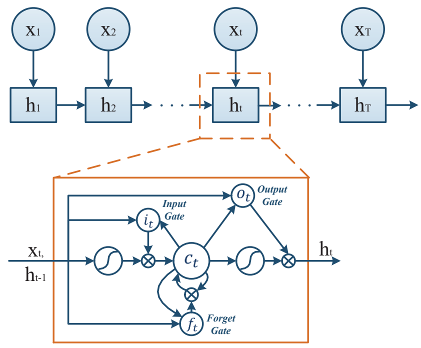

Fig. 2. The structure of RNN with LSTM units.

3 Recurrent Neural Network Models for Text Classification

So far, deep learning comes into play in many area, and achieves high performance [5, 6, 14, 8]. Many researchers have developed many neural network based sentence models [21, 18], which can be applied to conduct sentiment classification. In this paper, we adopt a recurrent neural network (RNN) with long short-term memory (LSTM) units [12] due to its great performance in handling multiple natural language processing (NLP) tasks [15].

3.1 Long Short-term Memory

LSTM is very effective in learning long-term dependencies. It has been proposed to address the issue that standard RNN suffers from severe gradients vanishing or exploding when dealing with long sequential data. The mathematical description of the LSTM structure is as follows:

$$\begin{bmatrix} \tilde{\mathbf{c}}_t \\ \mathbf{o}_t \\ \mathbf{i}_t \\ \mathbf{f}_t \end{bmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} \left(\mathbf{W}_p \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} + \mathbf{b}_p \right) \tag{1}$$

$$\mathbf{c}_t = \tilde{\mathbf{c}}_t \odot \mathbf{i}_t + \mathbf{c}_{t-1} \odot \mathbf{f}_t \tag{2}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right) \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^e$ is the input at the current time step, d denotes the number of the LSTM units, $\mathbf{W}_p \in \mathbb{R}^{4d\times(d+e)}$ and $\mathbf{b}_p \in \mathbb{R}^{4d}$ are parameters of affine transformation, σ denotes the logistic sigmoid function and $\odot$ denotes elementwise multiplication.

The update of each LSTM unit can be briefly summarized as follows:

$$\mathbf{h}_t = LSTM(\mathbf{h}_{t-1}, \mathbf{x}_t, \theta).$$

Function $LSTM$ is a combination of Equation(1) -(3), and θ represents all the parameters in the LSTM network. The structure of RNN with LSTM units is shown in Figure 2.

3.2 Text Classification with LSTM

Basically, for a given text sequence $x_t = \{x_1, x_2, ..., x_T\}$, the embedding layers[17][20] are used to find the representation vectors $\mathbf{x}_t$ for all words. Then, the representation vectors are input into the LSTM layers to output a representation vector $\mathbf{h}_T$. Finally, $\mathbf{h}_T$ is input into a fully connected layer to generate a probability distribution over all classes.

$$\hat{\mathbf{y}} = softmax\left(\mathbf{W}\mathbf{h}_T + \mathbf{b}\right)$$

where $\hat{\mathbf{y}} = \{\hat{\mathbf{y}}^1, \hat{\mathbf{y}}^2, ..., \hat{\mathbf{y}}^C\}$ represents the prediction probabilities for each class $j \in [1, C]$, $\mathbf{W}$ is the learned weights, and $\mathbf{b}$ is the bias.

For a given classic classification task, the loss function is defined as the cross-entropy between predicted and ground-truth distribution.

$$L\left(\hat{\mathbf{y}}, \mathbf{y}\right) = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j \log\left(\hat{y}_i^j\right) \tag{4}$$

where y_i^j is the ground-truth label for sample i regrading class j, N is the number of samples in the dataset, and C is the number of classes.

4 Bifurcated-LSTM for Cross-Domain Sentiment Classification

Motivated by multi-task learning, we design a Bifurcated-LSTM for cross-domain sentiment classification, which can divide some domain's reviews' feature space into domain-independent sentiment space and domain-dependent topic space.

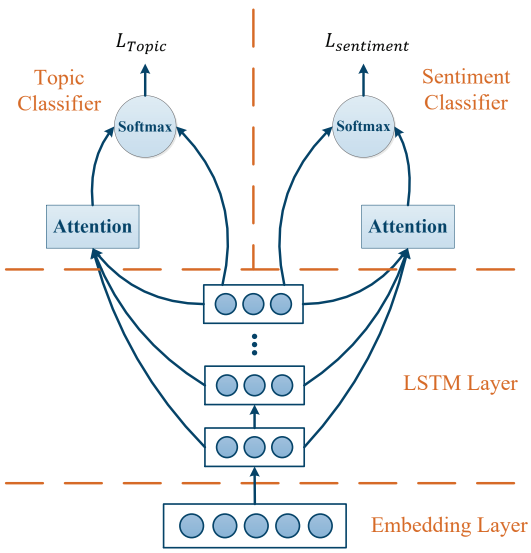

Fig. 3. The structure of Bifurcated-LSTM

The structure of a Bifurcated-LSTM is shown in Figure 3. First, a sentence passes through the embedding layer and LSTM layers to obtain the representation vector $\mathbf{h}_T$, which is the entire feature space of the text. Then, we simultaneously input $\mathbf{h}_T$ into two LSTM classifier branches, which have the same structure but for different objectives. One is for topic classification and the other is for sentiment classification. Topic features are needed to help the system distinguish source domain reviews from texts in other domains. To better achieve this, we augment the original dataset to obtain a more complete dataset for the system. Moreover, to accurately capture the features, we integrate the attention mechanism to the standard LSTM-based classifier for improving the categorization performance. In addition, to further enhance the performance of our model, we use orthogonal constraints strategy to separate the sentiment and topic features thoroughly. In the following, we describe dataset augmentation, Bifurcated-LSTM, and orthogonality constraints, respectively.

4.1 Dataset Augmentation

Our model aims at extracting topic-related and sentiment-related features from sentence representations. It is obvious that topic feature space varies in different domains. As a result, a model needs to be capable of obtaining distinct topic features from multiple domains.

Therefore, we reconstruct our training dataset by applying the dataset augmentation technique [8]. Specifically, we add some "noisy" data into the training dataset during the step of data collecting and preprocessing. These "noisy" data are text sequences picked from other domains, which have different topics from the ones in original dataset. After dataset augmentation, each data instance has two labels, and is denoted by (x, y^{Se}, y^{To}), where x is the text sequence, and y^{Se} is the sentiment label. $y^{To} \in \{0, 1\}$ is a binary label, where 1 indicates that the instance is from the current domain, and 0 means that the instance is a "noisy" sample.

4.2 Bifurcated-LSTM

As shown in Figure 3, the Bifurcated-LSTM is composed of the sentiment classifier, the topic classifier and the feature bifurcation. We describe them respectively in the following.

Attention-based LSTM Sentiment Classifier We integrate word embeddings and attention mechanism into the standard LSTM model to improve the performance of capturing the representative features from text sentences. Particularly, for a word x_t, we employ word embedding, like GloVe [20] and Word2Vec [17], to transform it into a representation vector $\mathbf{x_t}$. In addition, we adopt a word-level attention mechanism[1], which can identify the crucial part of a sentence, to improve the performance of our sentiment classifier.

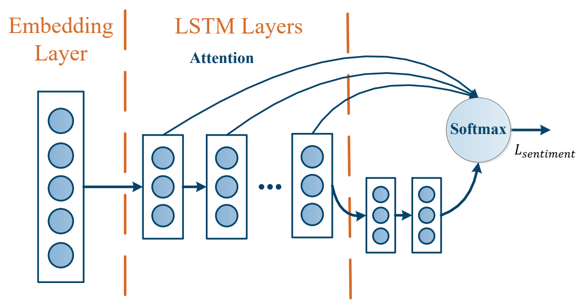

Fig. 4. Attention-based LSTM sentiment classifier.

As shown in Figure 4, in upper branch of the sentiment classifier, we apply attention mechanism at the common LSTM layers that are shared with the topic classifier, so that the information from the original training data can be further extracted and still exploited at the sentiment classifier. Let $\mathbf{H}_a \in \mathbb{R}^{d\times T}$ denote a matrix consisting of hidden vectors $[\mathbf{h}_1, ..., \mathbf{h}_T]$ produced by the LSTM, where d is the number of hidden layers and T is the length of a given sentence. The attention mechanism produces an attention weight vector $\mathbf{a}$ and a hidden representation $\mathbf{s}$ which is a weighted representation of a sentence with the given word. Both of them can be calculated as follows:

$$\begin{aligned}
\mathbf{a} &= softmax\left(\mathbf{w}^T \tanh(\mathbf{W}_h\mathbf{H}_a)\right) \\
\mathbf{s} &= \mathbf{H}\mathbf{a}^T
\end{aligned}$$

where we have $\mathbf{a} \in \mathbb{R}^T$, $\mathbf{s} \in \mathbb{R}^d$. $\mathbf{W}_h \in \mathbb{R}^{d\times d}$, and $\mathbf{w} \in \mathbb{R}^d$ are projection parameters.

In the lower branch of the sentiment classifier, particularly following common LSTM layers, we add several LSTM layers to extract sentiment features from the whole sentence feature space. The output of these LSTM layers is as follows:

$$\mathbf{h}_{output} = LSTM(\mathbf{h}_T, \mathbf{x}_t, \theta)$$

The final sentiment representation vector of the sentence, denotes by is given by:

$$\mathbf{h}^* = \tanh\left(\mathbf{W}_{Attention}\mathbf{s} + \mathbf{W}_{output}\mathbf{h}_{output}\right)$$

where $\mathbf{W}_{Attention}$ and $\mathbf{W}_{output}$ are projection parameters on the two branches of the sentiment classifier to be learned during the training process. Then, a $softmax$ layer is followed to transform $\mathbf{h}^*$ to the conditional probability distribution, i.e.,

$$\hat{\mathbf{y}} = softmax\left(\mathbf{W}_{softmax}\mathbf{h}^* + \mathbf{b}_{softmax}\right)$$

where $\mathbf{W}_{softmax}$ and $\mathbf{b}_{softmax}$ are the parameters for $softmax$ layer.

Based on Equation (4), the loss of sentiment classification can be computed as follows:

$$L_{Se}\left(\hat{\mathbf{y}}^{Se}, \mathbf{y}^{Se}\right) = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^{jSe} \log\left(\hat{y}_i^{jSe}\right) \tag{5}$$

where $\hat{\mathbf{y}}^{Se} = [\hat{\mathbf{y}}^{1Se}, \hat{\mathbf{y}}^{2Se}, ..., \hat{\mathbf{y}}^{jSe}]$ represents the predicted probabilities for each sentiment classification class $j \in [1, C]$, and $\mathbf{y}^{Se} = [\mathbf{y}^{1Se}, \mathbf{y}^{2Se}, ..., \mathbf{y}^{jSe}]$ represents the ground-truth labels, and N is the number of samples.

Attention-based LSTM Topic Classifier Note that both classifiers for topic and sentiment analysis share the same structure, and slightly difference lies in the objective function. Therefore, we simply show the loss function of the topic classifier as follows:

$$L_{To}\left(\hat{\mathbf{y}}^{To}, \mathbf{y}^{To}\right) = -\sum_{i=1}^{N'}\sum_{j=1}^{C'} y_i^{jTo} \log\left(\hat{y}_i^{jTo}\right) \tag{6}$$

where, similarly, $\hat{\mathbf{y}}^{To} = [\hat{\mathbf{y}}^{1To}, \hat{\mathbf{y}}^{2To}, ..., \hat{\mathbf{y}}^{jTo}]$ represents the predicted probabilities for each topic classification class $j \in [1, C']$, and $\mathbf{y}^{To} = [\mathbf{y}^{1To}, \mathbf{y}^{2To}, ..., \mathbf{y}^{jTo}]$ represents the ground-truth labels, and N' is the number of samples.

Feature Bifurcation The feature representation bifurcation is constructed by merging the sentiment classifier and the topic classifier. The shared attention-based LSTM layers condense an input sentence into a representation vector, which includes all features of the sentence. Each classifier only extracts the features that it is interested in according to the considered loss function.

4.3 Orthogonality Constraints

We notice that it is possible that the domain-dependent topic features and domain-independent sentiment features may interfere with each other. Inspired by recent work on multi-task learning [16] and shared-private latent space analysis [5], we employ the orthogonality constraint technique in our proposed feature divider. Specifically, it enables the divider to penalize commonly shared features in sentiment feature space and topic feature space and encourage to extract the independent sentiment topic features as purely as possible. To achieve this goal, we define the optimal loss function as follows:

$$L_{orth} = \sum_{i=1}^{N} \left\| \mathbf{H}_i^{Se\,\mathrm{T}} \mathbf{H}_k^{To} \right\|_F^2 \tag{7}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm, $\mathbf{H}^{Se}$ and $\mathbf{H}^{To}$ are two matrices whose rows are parameters from the private LSTM layers of sentiment classifier and topic classifier, respectively.

4.4 Training and Testing

Combining equation (5)-(7), the final loss function of our features divider model can be summarized as follows:

$$L = L_{Se} + L_{To} + \gamma L_{orth}$$

where γ is a hyperparameter.

In the training process, we feed the augmented dataset to the whole neural-based model to train the classifier. After training, we can obtain a Bifurcated-LSTM. For cross-domain sentiment classification task, we only focus on the sentiment classifier branch of the Bifurcated-LSTM. Therefore, in the testing process, we mainly transfer the well-trained sentiment classifier to other domains.

Table 1. Statistical knowledge of the 7 datasets. The columns 2-4 denote the number of samples in training, development and testing sets. The columns 5 and 6 represent the average length and vocabulary size of corresponding dataset.

Dataset	Train	Dev.	Test	Avg.L	Vocab.
Books	1400	200	400	159	62K
Electronics	1400	200	400	103	30K
DVD	1400	200	400	172	69K
Kitchen	1400	200	400	88	28K
Baby	1300	200	400	105	26K
Magazine	1300	200	400	113	30K
Software	1400	200	400	130	26K

5 Experiments Setting

5.1 Dataset

We collect product reviews of 7 different domains from Amazon [2]. First, we extract the comment sentences and corresponding labels from raw data, and then use keras [7] to perform the tokenization. After text preprocessing, we randomly partition all the datasets into a training set, a development set, and a testing set with the proportion of 70%, 10%, 20%, respectively. Table 1 shows the statistical information of all considered datasets.

5.2 Dataset Augmentation

In experiments, we randomly choose reviews from domains other than the considered source and target domains as "noisy" datase to conduct dataset augmentation. Meanwhile, we control the size of "noisy" dataset to be half size of the original training dataset. After combining the original and "noisy" training datasets, we have the augmented dataset.

5.3 Hyperparameters

We apply 200d GloVe vectors [20] to initialize the input sentence sequences, and $\gamma = 0.03$ in Eq. (8). Other parameters in the neural networks are initialized by randomly generated from a uniform distribution in [-0.1, 0.1]. We employ Adam to optimize our loss function shown in Eq. (8) with mini-batch size 24.

6 Performance Evaluation of Bifurcated-LSTM

6.1 Performance Evaluation

Table 2 shows the average error rate achieved by the proposed model, and compares it with that achieved by the one without domain adaptations. The LSTM networks in Bifurcated-LSTM are vanilla LSTM networks.

Table 2. Error rates of Bifurcated-LSTM for cross-domain classification. In "Bifurcated-LSTM" columns, the numbers in brackets represent the improvements relative to same domain classification results without domain adaptation.

Source Domain	Transferring to Target Domains without Domain Adaptation							Avg.
	Book	Elec.	DVD	Kitc.	Baby	Maga.	Soft.	
Book	20.8	21.3	22.7	23.2	23.0	23.3	21.7	22.29
Elec.	24.6	19.8	23.5	22.7	22.0	22.5	25.7	22.97
DVD	24.0	25.1	17.9	23.0	25.7	20.0	24.3	22.86
Kitc.	22.9	25.6	22.5	22.0	25.2	24.1	24.9	23.89
Baby	24.9	25.5	20.7	25.8	15.8	18.9	19.4	21.57
Maga.	24.4	23.0	24.6	21.3	21.6	11.2	18.1	20.60
Soft.	22.6	22.4	23.5	23.1	19.7	19.0	16.3	20.94
Source Domain	Transferring to Target Domains with Bifurcated-LSTM							Avg.
	Book	Elec.	DVD	Kitc.	Baby	Maga.	Soft.	
Book	17.6(-3.2)	18.3	19.1	19.4	19.2	18.1	19.0	18.67(-3.62)
Elec.	19.8	15.4(-4.4)	18.2	19.5	17.4	16.9	16.1	17.61(-5.36)
DVD	21.2	19.7	14.3(-3.6)	18.1	16.4	18.1	17.5	17.90(-4.96)
Kitc.	19.8	18.6	17.3	15.7(-6.3)	17.6	16.5	16.9	17.48(-6.41)
Baby	19.7	17.3	16.7	17.4	11.6(-4.2)	13.9	18.0	16.37(-5.20)
Maga.	20.1	16.2	16.9	18.1	14.3	6.9(-4.3)	17.1	15.65(-4.95)
Soft.	18.9	19.8	17.5	18.5	17.9	17.0	12.1(-4.2)	17.38(-3.56)

From this table, we can find that our proposed model can reduce the average error rate. Compared with the one without domain adaptation, our proposed model can reduce the error rate by 6.41%. Moreover, Table 2 also illustrates that our developed model can improve the performance of the classifier trained in its own domains, and the value can be up to 6.3%.

6.2 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 [2].
- **SFA**: Pan et al. proposed Spectral Feature Alignment (SFA) to build a bridge between source and target domains by aligning pivots with non-pivots [19].
- **DANN**: Ganin et al. applied the shallow version of Domain Adversarial Neural Networks (DANN) to the cross-domain sentiment classification [10].

We perform twelve domain adaptation tasks, and the results are in Table 3. We can find that our proposed model can achieve best performance on most tasks. For specific source domain, our proposed Bifurcated-LSTM always achieve the best average performance.

Table 3. Error rates of SCL, SFA, DANN, and Bifurcated-LSTM for cross-domain sentiment classification.

Source	Target	SCL	SFA	DANN	Bifurcated-LSTM
Kitc.	Book	33.9	25.2	29.1	**19.8**
Kitc.	Elec	16.3	**14.9**	15.7	18.6
Kitc.	DVD	24.6	23.0	26.0	**17.3**
Avg.		24.93	21.03	23.60	**18.57**
Book	Kitc.	21.3	21.2	22.1	**19.4**
Book	Elec.	22.5	27.5	26.7	**18.3**
Book	DVD	26.0	**18.6**	21.6	19.1
Avg.		23.27	22.43	23.47	**18.93**
Elec.	Kitc.	15.6	**13.3**	14.6	19.5
Elec.	Book	24.6	24.3	28.7	**19.8**
Elec.	DVD	25.7	22.8	26.2	**18.2**
Avg.		21.97	20.13	23.17	**19.17**
DVD	Kitc.	20.6	19.2	21.7	**18.1**
DVD	Book	23.2	22.5	27.7	**21.2**
DVD	Elec.	25.9	23.3	24.6	**19.7**
Avg.		23.23	21.67	24.67	**19.67**

7 Conclusion and Future Work

In this paper, we propose a Bifurcated-LSTM for cross-domain sentiment classification. In particular, this Bifurcated-LSTM can separate reviews' feature space into sentiment and topic feature subspaces. To enhance the performance of the Bifurcated-LSTM, we employ an attention mechanism to extract sentiment and topic features. Moreover, we also apply data augmentation and orthogonal constraints techniques to further improve the performance. We conduct extensive experiments to evaluate the performance of the proposed system.

Acknowledgement. This work was partially supported by the U.S. National Science Foundation under grants CNS-1602172 and CNS-1566479.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: ACL. vol. 7, pp. 440–447 (2007)
3. Bollegala, D., Mu, T., Goulermas, J.Y.: Cross-domain sentiment classification using sentiment sensitive embeddings. IEEE Transactions on Knowledge and Data Engineering 28(2), 398–410 (2016)
4. Bollegala, D., Weir, D., Carroll, J.: Cross-domain sentiment classification using a sentiment sensitive thesaurus. IEEE transactions on knowledge and data engineering 25(8), 1719–1731 (2013)

5. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: Advances in Neural Information Processing Systems. pp. 343–351 (2016)
6. Chen, X., Ji, J., Loparo, K., Li, P.: Real-time personalized cardiac arrhythmia detection and diagnosis: A cloud computing architecture. In: Biomedical & Health Informatics (BHI), 2017 IEEE EMBS International Conference on. pp. 201–204. IEEE (2017)
7. Chollet, F.: keras. `https://github.com/fchollet/keras` (2015)
8. DeVries, T., Taylor, G.W.: Dataset augmentation in feature space. arXiv preprint arXiv:1702.05538 (2017)
9. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: ACL (2). pp. 49–54 (2014)
10. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research 17(59), 1–35 (2016)
11. He, Y., Lin, C., Alani, H.: Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 123–131. Association for Computational Linguistics (2011)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
13. Li, T., Sindhwani, V., Ding, C., Zhang, Y.: Knowledge transformation for cross-domain sentiment classification. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 716–717. ACM (2009)
14. Liao, W., Salinas, S., Li, M., Li, P., Loparo, K.A.: Cascading failure attacks in the power system: a stochastic game perspective. IEEE Internet of Things Journal 4(6), 2247–2259 (2017)
15. Liu, P., Qiu, X., Chen, J., Huang, X.: Deep fusion lstms for text semantic matching. In: ACL (1) (2016)
16. Liu, P., Qiu, X., Huang, X.: Adversarial multi-task learning for text classification. arXiv preprint arXiv:1704.05742 (2017)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech. vol. 2, p. 3 (2010)
19. Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: Proceedings of the 19th international conference on World wide web. pp. 751–760. ACM (2010)
20. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
21. Socher, R., Pennington, J., Huang, E.H., Ng, A.Y., Manning, C.D.: Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the conference on empirical methods in natural language processing. pp. 151–161. Association for Computational Linguistics (2011)
22. Turney, P.D.: Thumbs up or thumbs down: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics. pp. 417–424. Association for Computational Linguistics (2002)