Low-Resource Cross-Domain Product Review Sentiment Classification Based on a CNN with an Auxiliary Large-Scale Corpus

Xiaocong Wei [1,2,*] iD **, Hongfei Lin** [1] **, Yuhai Yu** [1] iD **and Liang Yang** [1]

[1] School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China; hflin@dlut.edu.cn (H.L.); yuyh@dlnu.edu.cn (Y.Y.); yangliang@dlut.edu.cn (L.Y.)
[2] School of Software Engineering, Dalian University of Foreign Languages, Dalian 116044, China
[*] Correspondence: weixiaocong@dlufl.edu.cn; Tel.: +86-180-4010-7333

Abstract: The literature contains several reports evaluating the abilities of deep neural networks in text transfer learning. To our knowledge, however, there have been few efforts to fully realize the potential of deep neural networks in cross-domain product review sentiment classification. In this paper, we propose a two-layer convolutional neural network (CNN) for cross-domain product review sentiment classification (LM-CNN-LB). Transfer learning research into product review sentiment classification based on deep neural networks has been limited by the lack of a large-scale corpus; we sought to remedy this problem using a large-scale auxiliary cross-domain dataset collected from Amazon product reviews. Our proposed framework exhibits the dramatic transferability of deep neural networks for cross-domain product review sentiment classification and achieves state-of-the-art performance. The framework also outperforms complex engineered features used with a non-deep neural network method. The experiments demonstrate that introducing large-scale data from similar domains is an effective way to resolve the lack of training data. The LM-CNN-LB trained on the multi-source related domain dataset outperformed the one trained on a single similar domain.

1. Introduction

Much sentiment classification research focuses on training and testing classification models within a specific domain [1]. Transfer learning is a necessary technique to resolve the lack of labeled reviews in cross-domain product review sentiment classification. In many situations, plentiful product reviews from one domain (source domain) exist in which the sentiment is labeled but the application requires product review prediction from different, but related, domains (target domains) where the sentiment is unlabeled. For example, assume that the task is to build a book review classifier and that an insufficient number of labeled book reviews are available for training the classifier but that labeled electronics reviews are abundant. However, different domains have different feature spaces and distributions. In the book domain, words such as "exciting" and "graphic novel" are usually used to express a positive sentiment, whereas words such as "boring" or "drowsy" express a negative sentiment. In contrast, in the electronics domain, words such as "durable" and "light" are used to express a positive sentiment, whereas words such as "expensive" or "short battery life" often express a negative sentiment. Annotating reviews for a new domain is an expensive task that requires linguists skilled in natural language processing (NLP). Unlabeled reviews, however, are abundantly available. Nevertheless, when a sentiment classification model-trained source domain is directly applied to a different target domain, the result is often unsatisfactory because the bias between the source domain reviews and the target

domain reviews hinders the learning of an accurate sentiment classification model. In such scenarios, transfer learning focuses on the challenge of disentangling this domain discrepancy.

Researchers have proposed various methods to tackle this problem using non-deep neural networks [2–8]. However, these conventional approaches must extract customized features from the text and then feed the features into a classical shallow classifier such as a support vector machine (SVM). This process is empirical and task dependent, and it requires considerable engineering skills and domain expertise to construct features that are specific to a certain task. If a different classification task is addressed, this feature engineering method may not work well. However, such problems can be avoided if representative features can be automatically learned using a general-purpose learning procedure. This process is the key advantage of deep learning [9].

As a subfield of machine learning, deep learning aims to use a learning-based method to solve complex non-linear problems that normally rely on the processes and structure of the human brain. The deep learning approach can capture intermediate representations by constructing them hierarchically. Some research has been conducted to address various issues that text-fields face in deep neural networks. Kim [10] used word vectors as input and trained a simple CNN with a single convolutional layer to perform sentence classification. The word vectors were pre-trained by Word2Vec [11] on 100 billion words from Google News. The results suggested that pre-trained vectors are satisfactory and function as "universal" feature extractors. Fine-tuning the pre-trained vectors for each task produced further improvements. Kalchbrenner et al. [12] proposed a dynamic convolutional neural network that used dynamic K-max pooling to model sentences and introduced multiple feature maps that could capture short- or long-range semantic relations between words. This method did not rely on external sources, making the DCNN directly applicable to hard-to-parse sentences.

For transfer learning, deep neural networks can generalize well from one domain to another. The internal representation of the neural network contains no discriminative information about the raw input [13]. The transfer learning ability of a deep neural network in the image field has been evaluated [14,15]. Kandaswamy et al. [14] proposed a deep learning layer based on a feature transference approach for image classification. By transferring low-, middle- and high-layer features in unsupervised or supervised ways, using lower-layer features trained in a supervised fashion in the case of CNNs and unsupervised features trained in the case of SDAs, this method achieved superior performance. Yosinski et al. [15] quantified the transferability of each layer in an 8-layer convolutional neural network. For text, Pan et al. [16] proposed a multi-layer transfer learning method based on non-negative matrix tri-factorization. This method used supplementary latent factors to improve the transfer learning performance. It also built every layer by including common and specific latent feature spaces to reduce negative transfer. Collobert et al. [17] proposed a framework capable of multi-task transfer learning. Xiao Ding et al. [18] proposed a convolutional neural network method for mining user consumption intention (CIMM) on a child and baby corpus (source domain). This domain-adaptive framework was also effective for identifying user consumption intention in the movie domain (target domain). Studies have also focused on learning satisfactory representations via deep architecture to reduce transfer loss (improve the transfer ratio). Glorot et al. [19] proposed a deep learning method to learn an effective representation for domain adaptation based on stacked denoising auto-encoders. Bengio [20] proposed several challenging problems for transfer learning faces and investigated how to employ deep learning to resolve them. Mesnil et al. [21] found that models that interleaved different layer-wise representation learning algorithms performed well. Contractive auto-encoders, denoising auto-encoders and spike-and-slab RBMs worked best on dense datasets, and sparse rectifier denoising auto-encoders worked best on sparse datasets. Liu et al. [22] applied domain supervision and sentiment supervision to representation learning to address domain adaptation. By introducing domain labels and sentiment labels for loss functions based on KL divergence, the model could learn a more accurate domain-specialized and sentiment-specialized representation. The combination enabled the representation to be more domain-specific and sentiment-oriented and demonstrated that domain supervision is more effective than sentiment supervision. Gani et al. [23] proposed a representation

learning method by aligning the distributions of features across domains via standard back-propagation training. This method can be applied to both shallow and deep feed-forward architectures. To measure the transferability of neural network, similar to [15], which involved transfer learning in image analysis, Mou et al. [24] studied the transferability of semantic-relative and semantic-irrelative tasks, layers and parameter initialization, multi-task learning and their combinations on three datasets.

Other previous studies based on deep neural networks have focused on consumption intention [18], learning satisfactory representations [19–23], verifying the transferability of deep neural networks [24], multi-task transfer learning [17], online transfer learning [25]. Although there have been efforts to address product review text, such as [16], the neural network architecture and transfer schema used in this study are different from those presented in [16], and research on transferability still needs to be completed. No work has been conducted to fully realize the potential of deep neural networks in cross-domain product review sentiment classification, probably because existing product reviews for transfer learning have fewer resources. Transfer learning research of product review sentiment classification in deep neural networks has been severely limited by the lack of large-scale resources. The small size of benchmarks (e.g., 2000 reviews for each domain) [2] limits their utility as a research corpus. Deep learning methods typically outperform conventional non-deep neural networks on large-scale corpuses. Thus, this work remedies the problem of the lack of training data using an auxiliary large-scale cross-domain dataset collected from Amazon product reviews. Then, we demonstrate the transferability of the deep neural network using a cross-domain product review dataset. As a platform for evaluation, because the data are dramatically larger than any existing corpus of comparable quality, these data are suitable for training parameter-rich models such as deep neural networks, which have not previously been studied in this domain.

A CNN is a neural network that can learn the internal structure of data. CNNs have performed well in the image field; however, feeding a one-dimensional text data structure through the convolution layers leads to each unit in the convolution layer responding to only a small region of text. Existing studies on text with CNNs have attempted to resolve numerous problems, such as sentence modeling [12], relation classification [26,27], sentence level text classification [10], machine translation [28], short text classification [29], and domain-adaptive mining of user consumption intentions [18]. In this paper, a text transfer learning framework based on a two-layer convolutional neural network (LM-CNN-LB) is introduced that requires only a tiny number of labeled reviews from the target domain. Experiments over a large-scale auxiliary cross-domain dataset collected from Amazon product reviews demonstrate that the proposed framework can effectively learn a non-discriminative feature representation from the source domain and transfer it to the target domain.

The remainder of this paper is organized as follows: Section 2 describes the problem and presents definitions. Section 3 introduces the datasets used for the training method. The details of the neural network architecture are presented in Section 4. The results of a series of experiments to evaluate the effectiveness of the proposed solution are presented in Section 5. Section 6 concludes the work and outlines recommendations for future work.

2. Problem Setting

The definitions used in this work are presented below.

- **Domain**: A domain D consists of the following two components: a feature space, χ, and a marginal probability distribution, $P(X)$. χ is the space that includes all the term vectors, and X is an individual learning sample. In general, different domains have different feature spaces or different marginal probability distributions.
- **Source domain**: $D_S = \{(X_{S_i}, Y_{S_i})\}_{i=1}^{n_s}$ refers to a set of labeled reviews from a certain domain. X_{S_i} is the i-th labeled review, denoting one product review in the source domain. Y_{S_i} is the sentiment label of X_{S_i}, $Y_{S_i} \in \{+1, -1\}$, where the sentiment labels +1 and -1 denote positive and negative sentiments, respectively. n_S is the number of labeled instances in the source domain and denotes the total number of product reviews in the source domain.

- **Target domain**: $D_T = \{(X_{T_i})\}_{i=1}^{n_T}$ refers to a set of unlabeled reviews from a domain different from but related to the source domain. Here, X_{T_i} is the *i*-th unlabeled review corresponding to one product review in the target domain, and n_T is the number of unlabeled reviews in the target domain.

- **Cross-domain sentiment classification**: Cross-domain sentiment classification is defined as the task of training a binary classifier using labeled D_S to predict the sentiment label Y_{T_i} of a review X_{T_i} in the target domain.

3. Data Collection

Cross-domain product review transfer learning benchmark (D_b): A transfer learning benchmark defined as described by Blitzer et al. [2] has been widely used in many cross-domain sentiment classification methods. It contains Amazon product reviews consisting of the following four different product types: books (B_{D_b}), DVDs (D_{D_b}), electronics (E_{D_b}) and kitchen appliances (K_{D_b}). There are 1000 positive reviews and 1000 negative reviews for each domain. In this dataset, 12 pairs of cross-domain sentiment classification tasks are constructed as follows: $D_{D_b} \rightarrow B_{D_b}$, $E_{D_b} \rightarrow B_{D_b}$, $K_{D_b} \rightarrow B_{D_b}$, $K_{D_b} \rightarrow E_{D_b}$, $D_{D_b} \rightarrow E_{D_b}$, $B_{D_b} \rightarrow E_{D_b}$, $B_{D_b} \rightarrow D_{D_b}$, $K_{D_b} \rightarrow D_{D_b}$, $E_{D_b} \rightarrow D_{D_b}$, $B_{D_b} \rightarrow K_{D_b}$, $D_{D_b} \rightarrow K_{D_b}$, and $E_{D_b} \rightarrow K_{D_b}$, where the word before an arrow corresponds to the source domain and the word after an arrow corresponds to the target domain. Due to its small size, D_b is typically confined to conventional non-deep neural network methods and has been considered unsuitable for training parameter-rich neural networks.

Amazon product review dataset (D_l): To evaluate the ability of the deep neural network to perform cross-domain product review sentiment classification, a large-scale Amazon product review dataset collected by McAuley et al. [30] is introduced to supplement D_b. This dataset contains product reviews and scores from 24 product categories sold on Amazon.com, including 142.8 million reviews spanning from May 1996 to July 2014. Review scores lie on an integer scale from 1 to 5. Reviews with ratings of 1 and 2 are viewed as negative, and reviews with ratings of 4 and 5 are considered positive. For this study, the following four categories of original product reviews were extracted for testing purposes: Books (B_{D_l}), Movies and TV (D_{D_l}), Electronics (E_{D_l}) and Home and Kitchen (K_{D_l}) from the most recent year (2014). In total, 50,000 positive reviews and 50,000 negative reviews were used from each domain. Such a large-scale dataset can effectively reflect the transferability of a deep neural network.

Auxiliary product review dataset (D_{b_l}): KL (Kullback–Leibler) divergence is a method for measuring the similarity between two probability distributions and can be applied to measure the similarity between domains. Table 1 displays the KL divergence of the corresponding domain between D_b and D_l, while Table 2 shows the KL divergence between the different domains of D_b and D_l. In Table 2, the second and third columns show the KL divergence between the different domains in D_b and D_l, respectively. The fourth column is the KL divergence between the domains (the source domain is D_l and the target domain is D_b). Comparing Table 1 with Table 2, the KL divergence of the corresponding domains between D_b and D_l is far less than between the different domains in D_b or D_l. For example, the similarity of B_{D_b} and B_{D_l} is far higher than that of $E_{D_b} \rightarrow B_{D_b}$ or $E_{D_l} \rightarrow B_{D_l}$. Thus, a similar auxiliary corpus (denoted as D_{b_l}) can be constructed for each domain (see Table 3) to supplement the small-size benchmark D_b. The D_{b_l} corpus is sufficiently large to train parameter-rich models such as deep neural networks; thus, the use of deep neural networks is expected to succeed in terms of cross-domain product review sentiment classification and its evaluation on the benchmark dataset.

Table 1. Corresponding domain Kullback–Leibler (KL) divergence of D_b and D_l.

D_b	D_l	KL Divergence
B_{D_b}	B_{D_l}	0.1005126
D_{D_b}	D_{D_l}	0.0956735
E_{D_b}	E_{D_l}	0.0661170
K_{D_b}	K_{D_l}	0.0397648

Table 2. KL divergence of different domains.

Domain	D_l	D_b	$D_l{\rightarrow}D_b$
D→B	0.2643	0.2052	0.3017
E→B	0.6069	0.4864	0.5113
K→B	0.6048	0.5064	0.5126
B→D	0.2294	0.1925	0.2439
E→D	0.5225	0.4587	0.4734
K→D	0.5677	0.5043	0.4938
B→E	0.6153	0.3171	0.4014
D→E	0.5363	0.3025	0.3610
K→E	0.3010	0.2379	0.2189
B→K	0.6280	0.3029	0.3746
D→K	0.6211	0.2642	0.3425
E→K	0.3033	0.1936	0.1800

Table 3. Auxiliary dataset D_{bl} for cross-domain product review sentiment classification.

Domain	$D_b + D_l$	Positive	Negative
$B_{D_{bl}}$	$B_{D_b} + B_{D_l}$	50,000 + 1000	50,000 + 1000
$D_{D_{bl}}$	$D_{D_b} + D_{D_l}$	50,000 + 1000	50,000 + 1000
$E_{D_{bl}}$	$E_{D_b} + E_{D_l}$	50,000 + 1000	50,000 + 1000
$K_{D_{bl}}$	$K_{D_b} + K_{D_l}$	50,000 + 1000	50,000 + 1000

4. Neural Network Architecture

In this paper, a two-layer convolutional neural network is presented for cross-domain product review sentiment classification (**LM-CNN-LB**). This classification model is deployed with its convolution layers interleaved with pooling layers. The architecture of LM-CNN-LB is illustrated in Figure 1.

A CNN works only with fixed length inputs; thus, every input length is standardized to l by trimming the longer sentences and padding the shorter sentences with zeros. Given an input instance $X_i \in R^l$, x_i is the i-th word in this instance.

layer-0: The first layer is the embedding layer. Words in sentences are converted to low-dimensional word vectors $v_i \in R^k$ through pre-training by Word2Vec [11], where k is the dimension of the word vector. Thus, the input instance $X \in \mathbb{R}^{l \times k}$ is concatenated with the word vector as follows, where $\oplus$ is the concatenation operator:

$$X_{1:l} = v_1 \oplus v_2 \oplus ... \oplus v_1 \tag{1}$$

The word vectors are applied to initialize the weight of the embedding layer, and this layer is fine-tuned during training in the source domain. In subsequent steps, the network is regularized by dropout. The output can then be used to augment the neural network layer.

layer-1: This layer consists of a one-dimensional convolutional operation and a max pooling operation. The convolutional operation can learn an internal feature representation. The essence of the convolutional layer is to convert text regions of a fixed size (e.g., "do not waste your time" which has a size of 5) to feature vectors. The vector $\mathbf{m} \in R^n$ is the filter of the convolution, namely, the weight. A filter width of n enables the convolution layer to make use of text word order to capture the contextual feature of a word. For a word vector v_i, the feature vectors around v_i are concatenated within n. Then, the vector is input to the convolution operation to take the dot product of the weight vector $\mathbf{m} \in R^n$ and an input vector $v_i \in R^k$ with the activation function. The new feature representation is as follows:

$$f_j = \mathbf{m}^T v_{j:j+n-1} \tag{2}$$

The output of the layer is as follows:

$$\mathbf{O} = \sigma(f_j + b) \tag{3}$$

where σ is activation function *relu*.

$$f(x) = max(0, x) \tag{4}$$

The weight vector $\mathbf{m} \in R^n$ and the bias vector $b \in R^n$ are shared by all units in the same layer and are learned through training. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}\}$ to produce a feature map. By putting the one-dimensional max pooling layer on top of the local vectors, the network is enhanced, and it can capture the most useful local features for a task with a fixed size.
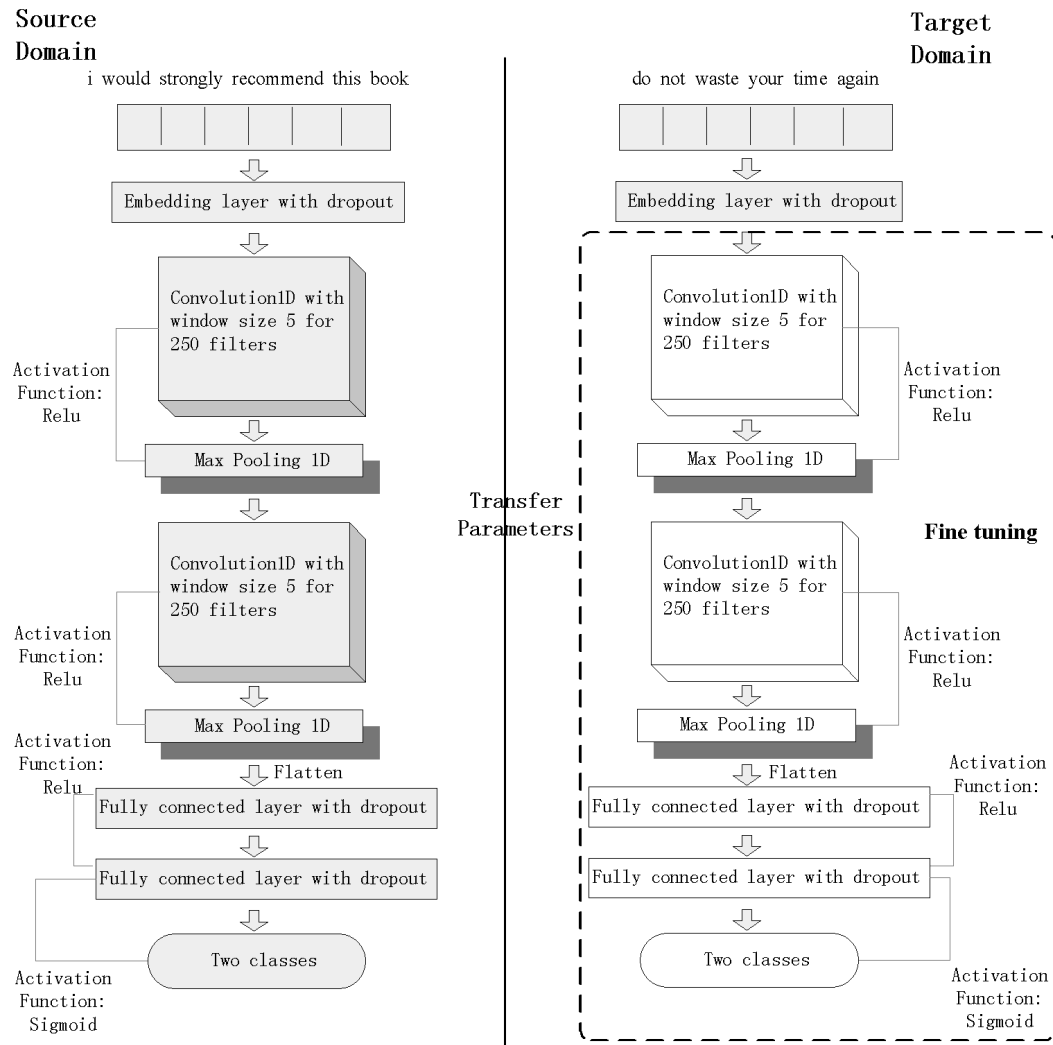

Figure 1. Architecture of LM-CNN-LB for cross-domain product review sentiment classification.

layer-2: The output of layer-1 is used as the input to this layer. Layer-2 is another layer consisting of a one-dimensional convolutional operation and a max pooling operation that is the same as layer-1.

layer-3: The output of layer-2 is then fed into a fully connected layer (regularized by dropout). The activation function is *relu*.

layer-4: The output of layer-3 is fed into a fully connected layer (regularized by dropout) at the end to classify the output. The activation function is *sigmoid*.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

In this scheme, a two-layer convolutional neural network is constructed and trained on the source domain data from D_{b_l}. The constructed model is designated as S_{cnn}. To transfer the data, another model (designated T_{cnn}) is built with the same neural network architecture as that of S_{cnn}. The parameters are transferred from S_{cnn} to initialize the corresponding layer in T_{cnn}. By freezing the embedding layer, the remaining layers of T_{cnn} are trained on a small quantity of target domain data from D_l to obtain the model T_{cnn} for the test target domain from D_b. For example, for task B→D, first, S_{cnn} is trained on the auxiliary dataset $B_{D_{bl}}$. Then, the parameters of S_{cnn} are transferred to T_{cnn} to initialize its neural network architecture. After freezing the embedding layer of T_{cnn}, the remaining layers are fine-tuned on small amounts of D_{D_l}. Finally, T_{cnn} is tested on D_{D_b}. Introducing the large-scale auxiliary data causes the small-size benchmark to become sufficient for training the neural network. Moreover, these conditions are suitable to fully demonstrate the transferability of deep neural networks. Furthermore, S_{cnn} can automatically extract feature representations that can be shared across the source and target domains. Finally, fine-tuning the parameters transferred from S_{cnn} using small amounts of target domain data enhances T_{cnn}, causing it to learn features that are specific to the target domain task and increasing its accuracy.

5. Experimental Evaluation

5.1. Benchmark Experiments

To evaluate the effectiveness of the proposed framework, the LM-CNN-LB method was compared with the following methods:

- SVM-NBB: A model trained on the source domain was directly applied to predict the target domain without any transfer learning method. The classifier was an SVM using the bag-of-words (BOW) schema and a linear kernel; the source domain and target domain were all from D_b. For example, the source domain was B_{D_b} and the target domain was D_{D_b}; the cross-domain classification task was $B_{D_b} \rightarrow D_{D_b}$.
- SVM-NLB: The model trained on the source domain was directly applied to predict the target domain without any transfer learning method. The classifier was SVM, using BOW and a linear kernel. The source domain was from D_{bl}, and the target domain was from D_b. For example, when the source domain was $B_{D_{bl}}$, the target domain was D_{D_b}, and the cross-domain classification task was $B_{D_{bl}} \rightarrow D_{D_b}$.
- SCL-MI: Blitzer et al. [2] applied structural correspondence learning for cross-domain sentiment analysis (SCL). SCL-MI was an improvement of SCL [31]. The source domain and the target domain were both from D_b.
- SFA: Spectral feature alignment was proposed by Pan et al. This approach bridged the gap between different source and target domains via word alignments [3]. The source domain and target domain were both from D_b.
- SS-FE: The SS-PE approach was used to conduct both labeling adaptation and instance adaptation for domain adaptation as in [5]. The source domain and target domain were each from D_b.
- CSC: The authors of [8] proposed a common subspace construction method for cross-domain sentiment classification called CSC. The source domain and target domain were each from D_b.
- PJNMF: This method links heterogeneous input features via pivots via joint non-negative matrix factorization [6]. The source domain and target domain were each from D_b.
- LM-CNN-NLB: LM-CNN-LB was applied to the source domain using D_{bl} to train the S_{cnn}. Then, the S_{cnn} was directly applied to predict the target domain from D_b.
- LM-CNN-BB: Like LM-CNN-LB but the S_{cnn} was trained on the source domain from D_b. The training set size was 1600 and the validation set size was 400. Then, the T_{cnn} was also trained on the target domain from D_b. The training set size was 400, the validation set size was 200, and the remaining 1400 data points comprised the test set.

In the process of training the S_{cnn} of LM-CNN-LB, the training set size was 90,000, and the validation set size was 12,000. To train the T_{cnn}, the training set size was 4000 (4% of the target domain from D_l) and the validation set size was 1000 (1% of target domain from D_l). Finally, the T_{cnn} model was used to test 2000 target domain product reviews from D_b. The embedding layer weights were pre-trained on Google News.

5.2. Experimental Configuration

The weight of the embedding layer in LM-CNN-LB was initialized with 300-dimensional word vectors. These vectors were pre-trained with 100 billion words from Google News or a self-built corpus by Google Word2Vec https://code.google.com/p/word2vec/. The training parameters are as follows: cbow is 0, size is 200, window is 8, negative is 25, hs is 0, sample is 1e-4, threads is 20, binary is 1 and iter is 15. For the self-built corpus, reviews from four domain, Books, Movies and TV, Electronics and Home and Kitchen were extracted from a large Amazon product review dataset covering the year 2014 (collected by McAuley et al. [30]). Neutral reviews (score rating 0) were removed, all the characters were converted to lower case, and punctuation was removed. The corpus contained 6.96 million product reviews, 386 million words and a vocabulary of 2.31 million words. Similarly, when preprocessing D_b and D_{bl}, punctuation was removed, and all characters were converted to lower case. For computational reasons, the review input lengths were normalized to 100. Accuracy was used as an evaluation metric, and the experiment was implemented as described by Keras 2.0 [32]. The programming language is Python and with NLTK natural language toolkit. The batch size of the neural network was 128. The number of convolution filters was 250, the filter width was 5, the convolution layer border mode was 'same', and the activation function was *relu*. The optimizer was *RMSProp*, the max pooling length was 2, and all dropouts were 0.1. The activation function of the first fully connected layer was *relu*, and the number of hidden units was 250. The activation function of the second fully connected layer was *sigmoid*, and the number of hidden units was 1. Shuffling occurred after every epoch. The training procedure periodically evaluated the binary cross-entropy objective function on the training set and the validation set. The test performance was associated with the last epoch validation accuracy. The learning rates were set to 0.0005. The S_{cnn} network was trained with 15 epochs, and the T_{cnn} network was trained with 50 epochs. Our networks are trained on one NVIDIA Tesla K20c GPU in a 64-bit Dell computer with two 2.40 GHz CPUs, 64 G main memories in Dalian, China, and Ubuntu 12.04. The runtime of training our framework (such as LM-CNN-BB) is: one epoch requires 123 s when training S_{cnn} on the kitchen appliances domain and 15 s when training T_{cnn} on the electronics domain.

Comparison results and discussion: Figure 2 shows the accuracy scores of the different methods for all pairs of tasks. The second set of bars shows that SVM-NLB performed better than SVM-NBB due to the introduction of the large-scale source domain. The large-scale similar auxiliary corpus was beneficial for cross-domain product sentiment classification tasks. Expanding the source domain data introduces more useful knowledge, which is the advantageous factor for cross-domain classification, is introduced. However, SVM-NLB still performs worse than methods that applied the transfer learning method, such as SCL-MI, SFA and SS-FE. Although SVM-NLB can learn more knowledge from large-scale data for cross-domain classification tasks, this algorithm is insufficient to bridge the gap between different domains. However, when LM-CNN-NLB is applied to the same corpus as SVM-NLB, the performance improves greatly. This improvement can be attributed primarily to the following two factors: first, as is well-known, word embedding is more effective than BOW. BOW is unable to capture the complex linguistic phenomena of words that are available in the word embedding representation, in which semantically close words are likewise close in the lower-dimensional vector space. Thus, word embedding contains more semantic information. The results confirm that the pre-trained vectors are both satisfactory and "universal" feature extractors that are beneficial for cross-domain classification. Second, through the convolutional and max pooling operations, LM-CNN-NLB can effectively use local contextual features and global contextual features to capture the generic variations present in all

factors that are suitable for cross-domain classification. LM-CNN-NLB behaves better than previous baselines on most tasks, while LM-CNN-LB achieves the best performance. When training S_{cnn} on the source domain, it extracts no discriminative feature representation, therefore, the features can be shared across the source domain and target domain. Fine-tuning the parameters transferred from S_{cnn} on a small amount of target domain data causes the T_{cnn} to learn a feature specific to the target domain task. Comparing LM-CNN-BB with LM-CNN-LB, when the available corpus is small, it is inappropriate for training parameter-rich models such as neural networks. However, introducing a large-scale similar auxiliary dataset effectively expands the corpus. By transferring parameters from the convolutional neural network trained on the source domain to initialize another identical neural architecture and then fine-tuning with small amounts of target domain data, the target network can learn a feature representation that generalizes well across different domains.
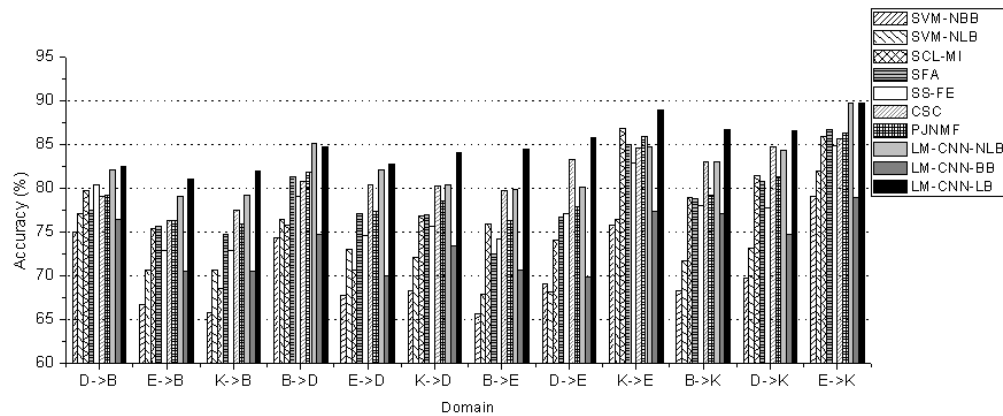

Figure 2. Comparison of different methods on the benchmark dataset for cross-domain sentiment classification.

Corpus size and multi-source domain: Figure 3 shows the effect of corpus size and a multi-source domain. For a multi-source domain, one domain was used as the target domain, and the other three domains were used as the source domain. The three source domains were related but were different from the target domain. For computational reasons, each domain dataset consisted of 20,000 reviews extracted from D_l and 2000 from D_b. For example, the S_{cnn} was trained on source domain reviews consisting of the following three domains: $B_{D_{bl}}$, $D_{D_{bl}}$ and $E_{D_{bl}}$. To train the S_{cnn}, the source domain dataset was split into a training set and a validation set. The training set included 54,000 reviews (90% of each domain extracted from D_l) and a validation set of 12,000 reviews. The T_{cnn} was trained on the target domain K_{D_l} using a training set size of 700 and a validation set size of 300, totalling 5% of K_{D_l}. The T_{cnn} model can then be used to predict K_{D_b} (denoted as the multi-source domain in Figure 3). The multi-source domain was also compared with LM-CNN-LB on different corpus sizes. LM-CNN-LB62k andLM-CNN-LB22k were the datasets for each domain consisting of 60,000 data points, 20,000 reviews extracted from D_l and 2000 from D_b. The source domain dataset for training the S_{cnn} from D_{bl} was split into a training set and a validation set. The training set was 90% of each domain extracted from D_l. The validation set consisted of the remaining reviews from each domain extracted from D_l plus 2000 from D_b. To train T_{cnn} on the target domain, the training set and the validation set totaled 5% of each domain extracted from D_l. The compositions of the corpuses are displayed in detail in Table 4. The accuracy of the target domain dataset is the average score of the other domains used for prediction. For example, for LM-CNN-LB62k, accuracy of D→B, E→B, K→B is $Acur_D$, $Acur_E$ and $Acur_K$, respectively, while the average classification accuracy for Books is $(Acur_D + Acur_E + Acur_K)/3$. The embedding layer weight of the above methods is a 300-dimension word vector pre-trained on the self-built corpus described in Section 5.2.
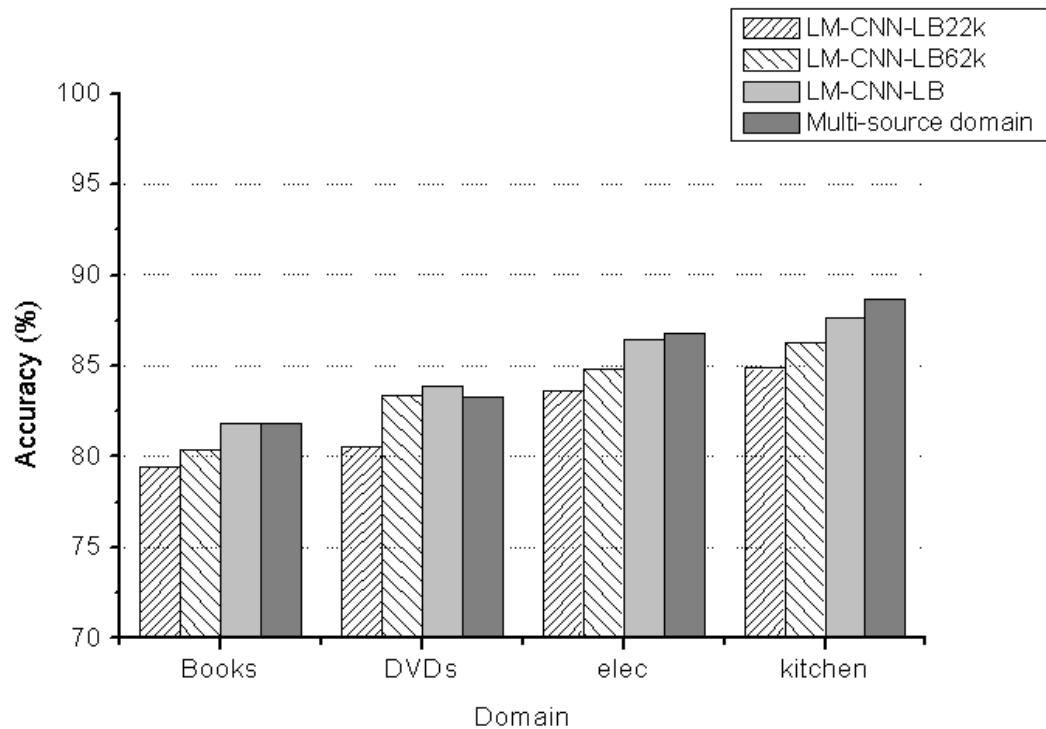
Figure 3. Effect of corpus size and multi-source domain.

As shown in Figure 3, the proposed method performed better using the larger corpus. This finding demonstrates that a large corpus size boosts accuracy. However, the best performance was achieved by a multi-source-related domain decoupled from the corpus size; the proportions of the training and validation set used for T_{cnn} are both smaller than those used for LM-CNN-LB. The results indicate that transfer parameters trained on multi-source related domain data yield superior performance. Based on multi-source related domain data, a convolutional neural network can discover features that capture the generic variations across a wide range of factors. In other words, training with multi-source-related domain data further promotes cross-domain classification.

Table 4. Composition of different size corpus.

Method	Training (S_{cnn})	Validation (S_{cnn})	Training (T_{cnn})	Validation (T_{cnn})
LM-CNN-LB22k	18,000	4000	700	300k
LM-CNN-LB62k	54,000	8000	2000	1000
LM-CNN-LB	90,000	12,000	4000	1000
Multi-source domain	54,000	12,000	700	300

Corpus for training word vector: Using LM-CNN-BB, the word embedding weight pre-trained on Google News and the self-built corpus were also compared. Figure 4 shows that the word vector pre-trained on a self-built corpus yields a superior performance over one pre-trained on Google News in all cross-domain tasks. It can be concluded that a pre-trained word vector from a corpus collected specifically for a certain task performs better than on pre-trained on Google News in cross-domain product review sentiment classification.
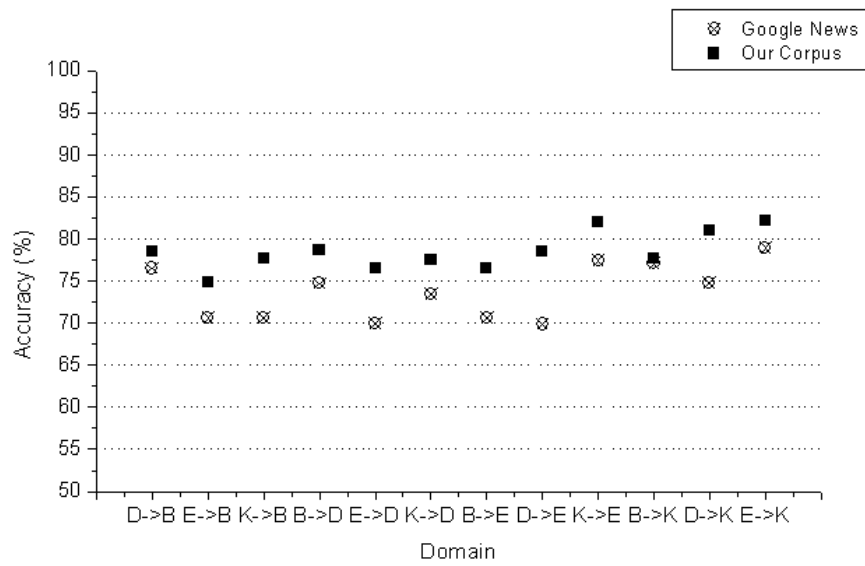
Figure 4. Comparison of word embedding pre-trained on Google News and the proposed corpus.

5.3. Large-Scale Corpus Experiments

The transferability was exhibited on the large-scale dataset D_l, as shown in Figure 5. SVM-NLL was the same as SVM-NLB, but the target domain test set was from D_l. LM-CNN-LL22k, LM-CNN-LL62k, and LM-CNN-LL were the same as LM-CNN-LB22k, LM-CNN-62k and LM-CNN-LB, but the T_{cnn} model was used to test the reviews from D_l (excluding those used for training and validation). The results shown in Figure 5 indicate that the proposed method performed better than SVM-NLL. A larger corpus results in higher accuracy; this result is consistent with experiments on the benchmark D_b.
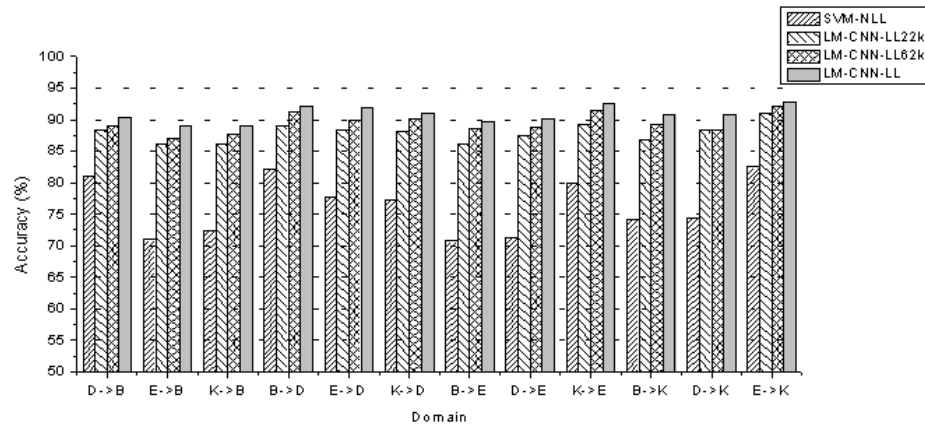

Figure 5. Comparison of different methods and corpus size on D_l for cross-domain sentiment classification.

Evaluating transfer readiness: In this part, the transferability of LM-CNN-LL62k is evaluated by freezing different layers of T_{cnn}, as shown in Table 5. For example, layer-0 of T_{cnn} was frozen after transferring parameters from S_{cnn} to T_{cnn}. The parameters of layer-0 were constant during training. The parameters of the remaining layers (layer-1, layer-2, layer-3, and layer-4) were trained and fine-tuned using small amounts of labeled reviews from the target domain. The frozen layer/s of T_{cnn} are listed in the first column of Table 5, and the test accuracy is listed in the second column.

Fine-tuning more layers stimulates learning features that are specific to the target domain. The more layers that participate in fine-tuning, the more specific the features are to the target domain.

Moreover, we analyzed the effect of epoch number when training the S_{cnn}. As illustrated in Figure 6, the accuracy increases from epochs 1 to 4. Thus, a larger number of epochs does not necessarily lead to higher performance, suggesting that well-trained parameters from the source domain are crucial for this work. The selected number of epochs causes the S_{cnn} to learn discriminative feature representation that initializes T_{cnn} well. However, if the source domain is excessively trained, the parameters transferred from the S_{cnn} to the T_{cnn} will be overfit to the source domain. The learned feature representation of the S_{cnn} is then excessively specific to the source domain but will be underfit to the target domain.

Table 5. Main results of LM-CNN-LL62k for freezing different layers.

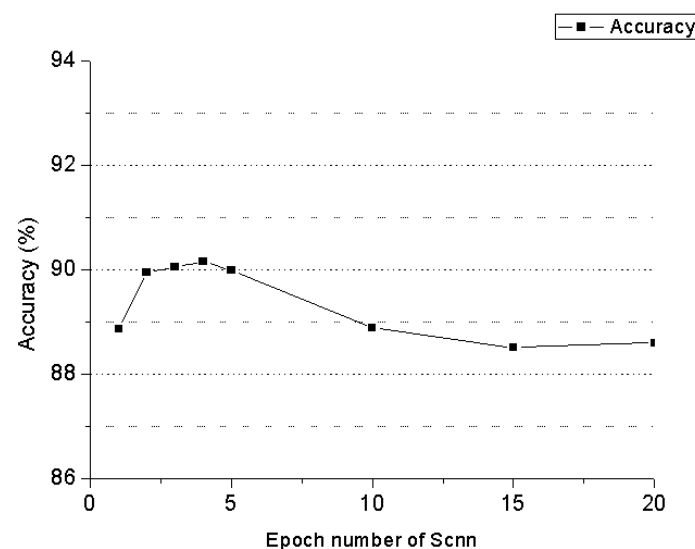| Frozen Layer/s	Accuracy
0,1,2,3	88.12
0,1,2	88.13
0,1	88.66
0	89.42

Figure 6. Effect of S_{cnn} epoch number.

Labeled target domain data: We also tested the effect of using different proportions of the labeled target domain data, which varied from 1/120 to 1/5 for every domain of LM-CNN-LL62k. Figure 7 indicates that a larger number of labeled target domain data results in higher test accuracy in every task. It is not surprising that LM-CNN-LL62k can learn more information specific to the target domain from the extensive labeled data. The labeled data of the target domain is crucial for learning an effective feature representation for cross-domain sentiment classification.
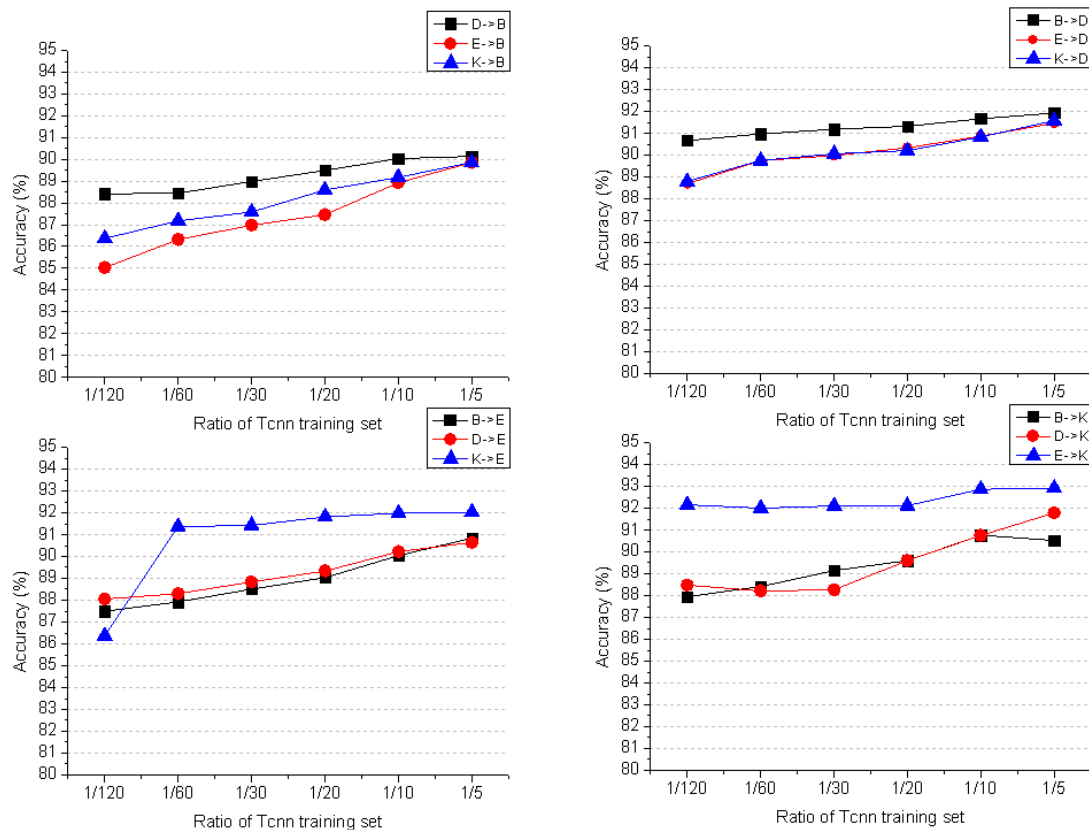
Figure 7. Effect of labeled target domain data ratio for training the T_{cnn}.

6. Conclusions

This paper proposes a two-layer convolutional neural network for cross-domain product review sentiment classification. The available low-resource benchmark dataset was unsuitable for training a deep neural network; thus, a large-scale auxiliary corpus was introduced. The proposed scheme exhibits the potential of deep neural networks for cross-domain product review sentiment classification. This increase in scale allows product review sentiment classification to outperform sophisticated existing methods (regardless of whether they are based on a deep neural network). The use of a large-scale corpus allowed a neural-network-based model to perform competitively on the product review sentiment classification benchmark for the first time. These experimental results demonstrate that introducing a large-scale corpus from a similar domain can significantly boost accuracy. The larger the corpus, the higher the accuracy achieved. Moreover, parameters transferred from a multi-source related domain are more effective than those transferred from a single similar domain. Freezing layers is detrimental to transfer performance, and choosing appropriate epochs for training source domain results improves performance. Introducing additional labeled data from the target domain for fine-tuning leads to better transferability.

Research on text transfer learning based on deep neural networks is currently in its infancy, and more efforts should be made to improve the learning algorithms. This paper describes new state-of-the-art results achieved while requiring only small amounts of labeled target domain data. A more complete comparison of approaches based on deep neural networks will be addressed in future work. Further studies on cross-domain product review sentiment classification on small datasets with deep neural networks are also essential for future study.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN   Convolutional neural network
NLP   Natural language processing

References

1. Recupero, D.R.; Presutti, V.; Consoli, S.; Gangemi, A.; Nuzzolese, A.G. Sentilo: Frame-based sentiment analysis. *Cognit. Comput.* **2015**, *7*, 211–225.
2. Blitzer, J.; McDonald, R.; Pereira, F. Domain adaptation with structural correspondence learning. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP), Sydney, Australia, 22–23 July 2006; pp. 120–128.
3. Pan, S.J.; Ni, X.; Sun, J.T.; Yang, Q.; Chen, Z. Cross-domain sentiment classification via spectral feature alignment. In Proceedings of theWorldWideWeb (WWW), Raleigh, NC, USA, 26–30 April 2010; pp. 751–760.
4. Bollegala, D.; Weir, D.; Carroll, J. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 1719–1731.
5. Xia, R.; Zong, C.; Hu, X.; Cambria, E. Feature ensemble plus sample selection: Domain adaptation for sentiment classification. *IEEE Intell. Syst.* **2013**, *28*, 10–18.
6. Zhou, G.; He, T.; Wu, W.; Hu, X.T. Linking heterogeneous input features with pivots for domain adaptation. In Proceedings of the International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 1419–1425.
7. Li, S.; Xue, Y.; Wang, Z.; Zhou, G. Active learning for cross-domain sentiment classification. In Proceedings of the International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 2127–2133.
8. Zhang, Y.; Xu, X.; Hu, X. A common subspace construction method in cross-domain sentiment classification. In Proceedings of the Conference on Electronic Science and Automation Control, Zhengzhou, China, 15–16 August 2015; pp. 48–52.
9. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
10. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the Empirical Methods on Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
11. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
12. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. In Proceedings of the Association for Computational Linguistics (ACL), Baltimore, MD, USA, 22–27 June 2014; pp. 655–665.
13. Lu, J.; Behbood, V.; Hao, P.; Xue, S.; Zhang, G. Transfer learning using computational intelligence: A survey. *Knowl.-Based Syst.* **2015**, *80*, 14–23.
14. Kandaswamy, C.; Silva, L.M.; Alexandre, L.A.; Santos, J.M.; de Sá, J.M. Improving deep neural network performance by reusing features trained with transductive transference. In Proceedings of the International Conference on Artificial Neural Networks (ICANN), Hamburg, Germany, 15–19 September 2014; pp. 265–272.
15. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
16. Pan, J.; Hu, X.; Li, P.; Li, H.; Li, W.; He, Y.; Zhang, Y.; Lin, Y. Domain adaptation via multi-layer transfer learning. *Neurocomputing* **2016**, *190*, 10–24.

17. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.

18. Ding, X.; Liu, T.; Duan, J.; Nie, J.Y. Mining user consumption intention from social media using domain adaptive convolutional neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2389–2395.

19. Glorot, X.; Bordes, A.; Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 513–520.

20. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In Proceedings of the International Conference on Unsupervised and Transfer Learning Workshop, Edinburgh, UK, 26 June–1 July 2012; pp. 17–36.

21. Mesnil, G.; Dauphin, Y.; Glorot, X.; Rifai, S.; Bengio, Y.; Goodfellow, I.J.; Lavoie, E.; Muller, X.; Desjardins, G.; Warde-Farley, D. Unsupervised and Transfer Learning Challenge: a Deep Learning Approach. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, USA, 27 June–1 July 2012; pp. 97–110.

22. Liu, B.; Huang, M.; Sun, J.; Zhu, X. Incorporating domain and sentiment supervision in representation learning for domain adaptation. In Proceedings of the International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 1277–1283.

23. Gani, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2015**, *17*, 1–35.

24. Mou, L.; Meng, Z.; Yan, R.; Li, G.; Xu, Y.; Zhang, L.; Jin, Z. How Transferable are Neural Networks in NLP Applications? In Proceedings of the EMNLP, Austin, TX, USA, 1–4 November 2016; pp. 479–489.

25. Seera, M.; Lim, C.P. Transfer learning using the online fuzzy min-max neural network. *Comput. Appl.* **2014**, *25*, 469–480.

26. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the International Conference on Computational Linguistic (COLING), Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.

27. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from con-volutional neural networks. In Proceedings of the VS@HLT-NAACL, Denver, CO, USA, 31 May–5 June 2015; pp. 39–48.

28. Meng, F.; Lu, Z.; Wang, M.; Li, H.; Jiang, W.; Liu, Q. Encoding source language with convolutional neural network for machine translation. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Beijing, China, 26–31 July 2015; pp. 20–30.

29. Dos Santos, C.N.; Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the International Conference on Computational Linguistics (COLING), Dublin, Ireland, 23–29 August 2014; pp. 69–78.

30. McAuley, J.; Pandey, R.; Leskovec, J. Inferring networks of substitutable and complementary products. In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'15), Sydney, Australia, 10–13 August 2015; pp. 785–794.

31. Blitzer, J.; Dredze, M.; Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech Republic, 23–30 June 2007; pp. 440–447.

32. Chollet, F. Keras. Available online: http://github.com/fchollet/keras (accessed on 19 July 2017).