Intelligent Data Analysis 23 (2019) 779–797
DOI 10.3233/IDA-184131
IOS Press

Learning of indiscriminate distributions of document embeddings for domain adaptation

Saerom Park[a], Woojin Lee[b] and Jaewook Lee[b,*]
[a]*Industrial and Mathematical Data Analytics Research Center, Seoul National University, Seoul, Korea*
[b]*Industrial Engineering, Seoul National University, Seoul, Korea*

Abstract. Natural language processing (NLP) is an important application area in domain adaptation because properties of texts depend on their corpus. However, a textual input is not fundamentally represented as the numerical vector. Many domain adaptation methods for NLP have been developed on the basis of numerical representations of texts instead of textual inputs. Thus, we develop a distributed representation learning method of words and documents for domain adaptation. The developed method addresses the domain separation problem of document embeddings from different domains, that is, the supports of the embeddings are separable across domains and the distributions of the embeddings are discriminated. We propose a new method based on negative sampling. The proposed method learns document embeddings by assuming that a noise distribution is dependent on a domain. The proposed method moves a document embedding close to the embeddings of the important words in the document and keeps the embedding away from the word embeddings that occur frequently in both domains. For Amazon reviews, we verified that the proposed method outperformed other representation methods in terms of indiscriminability of the distributions of the document embeddings through experiments such as visualizing them and calculating a proxy A-distance measure. We also performed sentiment classification tasks to validate the effectiveness of document embeddings. The proposed method achieved consistently better results than other methods. In addition, we applied the learned document embeddings to the domain adversarial neural network method, which is a popular deep learning-based domain adaptation model. The proposed method obtained not only better performance on most datasets but also more stable convergences for all datasets than the other methods. Therefore, the proposed method are applicable to other domain adaptation methods for NLP using numerical representations of documents or words.

Keywords: Domain adaptation, natural language processing, distributed representation, negative sampling

1. Introduction

Domain adaptation assumes that the distributions of training (source) and test (target) data are similar but different unlike the assumption in traditional machine learning. Domain adaptation approaches construct a predictive model across domains (source and target domains) by learning common features [11,13], weighting instances [14,16,23], or matching training data to target data [6]. An important issue in the domain adaptation problem is obtaining good representation because the difference in the distributions of the inputs fundamentally originates from their representations. Therefore, learning representations for domain adaptation is an interesting topic.

Representation learning methods for domain adaptation can be divided into semi-supervised and unsupervised methods according to whether target data are partially labeled or unlabeled. Semi-supervised

*Corresponding author: Jaewook Lee, Industrial Engineering, Seoul National University, 1 Gwanakro, Gwanak-gu, Seoul 08826, Korea. E-mail: jaewook@snu.ac.kr.

methods use partially labeled target data for learning newly transformed features from the original representations and training classifiers [9,24,25]. Many deep learning methods are developed recently to simultaneously learn shared representations and predictive models; these methods achieve state-of-the-art performance despite using only source labels [7,10,19,26]. These unsupervised domain adaptation methods use source labels, but some researchers have developed autoencoder-based methods that learn shared representations even without source labels [8,11]. These methods train stacked denoising autoencoder (sDAE) or marginalized stacked denoising autoencoder (mSDA) with source and target data without label information and obtain effective and generic features, which are invariant across domains in natural language processing (NLP) tasks. However, the performance of these models is inherently dependent on the initial representations of documents because they start from the numerical representations of texts instead of from textual inputs.

Many domain adaptation applications for NLP tasks, such as sentiment analysis or part of speech (PoS) tagging, have been presented [4,6,11,24]. Domain adaptation strategies can be appropriately implemented in NLP applications because text documents exhibit different distributions for various domains. However, the numerical representations of words or documents should be obtained before applying domain adaptation methods because the text input is not originally located in a real-valued vector space. Many researchers have used dictionary-based representations, such as bag of words (BoW), bag of n-grams, and term frequency inverse document frequency (TF-IDF) [3,4,6]. Bollegala et al. [6] obtained word embeddings by separating pivot and non-pivot words and learned document embeddings through word embeddings and the sentiment classifier; however, these processes require high training costs. The structural correspondence learning (SCL) method also separates pivot and non-pivot features, learns shared feature representation by introducing auxiliary pivot prediction problems, and improves the performance of PoS tagging and sentiment classification tasks [3,4]. However, these methods require source labels, design an auxiliary prediction problem [3,4], or construct a neighborhood graph [6] to learn document or word representations from unigram or bigram features.

Distributed representation models of words and documents have been proposed and effectively applied to NLP tasks, wherein they exhibit superior performance to the dictionary-based representation model that loses the order information of words [18,20,21]. Distributed representation models can capture meaningful relationships between documents or words effectively without sparsity and high dimensionality problems. However, the distributed representations are unsuitable to domain adaptation problems because different domains have dissimilar distributions of words and documents. Bollegala et al. [5] proposed an unsupervised crossdomain word representation method, which learns domain-specific word embeddings and pivot word embeddings, based on distributed representation learning methods. However, this method focuses only on learning word embeddings; therefore, document embeddings are not learned but obtained by aggregating the included word embeddings, wherein source labels are also used to learn the aggregation weights.

In the present study, we aim to develop a distributed representation learning method of words and documents for domain adaptation. We expect that this method can capture semantic relationships, reduce the difference of the distributions of embeddings in source and target domains, and provide the base document representation for other domain adaptation prediction models. The proposed method can learn generic and effective embeddings from the original textual inputs without any label information.

The remainder of the paper is organized as follows. In Section 2, we review representation learning methods for domain adaptation and distributed representation of words and documents. In Section 3, we propose domain-adapted distributed representation methods of words and documents with the illustrative example of word embeddings. In Section 4, we demonstrate the effectiveness of the proposed methods by comparing them with other document representation methods in terms of visualization, classification, and application to other domain adaptation models. Finally, Section 5 presents the conclusion.

2. Related work

We first review the representation learning methods for achieving domain adaptation in NLP tasks, and then we briefly introduce distributed representation learning of words and documents.

2.1. Representation learning for domain adaptation

A model trained with source data is difficult to generalize to the target data when source and target distributions differ. In NLP applications, distribution depends on the representation of documents or words as numerical vectors, which indicates that learning representation is important in the domain adaptation problem. Therefore, most domain adaptation models aim to find new features that minimize the difference of the numerical representations of words or documents between the source and target. Glorot et al. [11] used BoW representations as input and trained sDAE by layers to obtain hidden representations that can characterize documents across domains. Chen et al. learned new representations by using mSDA and overcame the limitations in [11], such as high computational cost and high dimensionality problem caused by dictionary-based representation [8]. Although this model is effective in domain adaptation for sentiment classification, it is still dependent on the initial representations because it requires the numerical representation of documents as an input [8].

State-of-the-art models that use deep learning techniques directly minimize domain divergence between domain-specific features and classification loss of representations in the source domain [7,10,26]. These models are applicable to all application fields in which numerical vector representations are provided. Therefore, they also use BoW representations to represent textual input as numerical vectors when the sentiment classification task is given. Ganin et al. developed the domain adversarial neural network (DANN) algorithm that applies the concept of adversarial training to extract common features from the source and target domains [10]. The extracted features should be discriminative to a given classification task and indiscriminate between the source and target domains; this condition can be achieved using the gradient reversal layer in the domain classifier. The DANN algorithm minimizes H-divergence [1], which measures the distance between the source and target distributions. The DANN model trains new features from BoW and mSDA representations for a sentiment analysis dataset [10]. As shown in the result, DANN on mSDA demonstrates considerably superior performance to the original (BoW) representations, which implies that an effective representation can affect the performance of domain adaptation. Therefore, we use our document representations as the input feature of DANN and train the DANN model to identify the effectiveness of our representations. We aim to develop an improved document representation that can replace dictionary-based representation in NLP domain adaptation tasks. Distributed representation methods for words and documents have outperformed dictionary-based methods in many NLP tasks; thus, we introduce these methods in the following section before proposing a new method based on these approaches.

2.2. Distributed representation of words and documents

In distributed representation, similar instances are close in the representation space. The definition of "similar" depends on the application field. In the case of text documents, "similar" words have similar context or occur near one another, and "similar" documents contain "similar" words. From such perspectives, the distributed representation model of words (the Word2vec model) learns word embeddings to increase the probability of its context words given a word (the skip-gram model) or the probability of a word given its context words (the continuous BoW (CBoW) model). Consequently, close word vectors

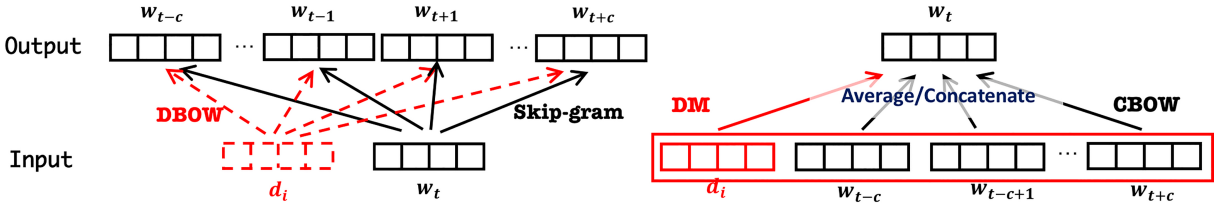


Fig. 1. Architecture of distributed models: the skip-gram and CBoW models learn word embeddings, whereas the DBoW and DM models learn document embeddings.

exhibit similar semantic and syntactic characterizations. On the basis of these models, Le and Mikolov developed the distributed representation learning model of sentences and documents (Doc2Vec); this model learns the representations of a document by maximizing the probability of the words in the document given the document and the context words (the distributed memory (DM) model) or the probability of the words in the document given the document (the distributed BoW (DBoW) model) [18]. Figure 1 illustrates the architecture of the aforementioned distributed models.

Distributed representation models learn input and output embeddings by maximizing the following conditional probability:

$$p_\theta(O|I) = \frac{\exp\left(s_\theta(O,I)\right)}{\sum_{j\in\mathscr{O}} \exp\left(s_\theta(j,I)\right)} \tag{1}$$

where v_i is an input embedding of instance i, $\tilde{v}_j$ is an output embedding of instance j, $s_\theta(j,i) = \tilde{v}_j^\top v_i + b_j$ is a score, and $\mathscr{O}$ is the set of all possible outputs. For example, in the skip-gram model, the conditional independence of the context words given a word w_t is assumed. Hence, the objective function is given by:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j\in\mathscr{I}} \log p(w_{t+j}|w_t), \tag{2}$$

where T is the length of word sequence $w_1,\ldots,w_T$, c is the size of the contexts, and $\mathscr{I} = \{-c, \ldots,-1,1,\ldots,c\}$. In the DBoW model, the conditional probability of words that comprise a document when the document is maximized similarly to the skip-gram model is given in Eq. (2). However, Eq. (1) is impractical for optimization because of the calculation normalization constant of all possible outputs. Distributed representation models considerably reduce this computational cost by subsampling frequent words [21] and using noise contrastive estimation (NCE) [15,22]. These models outperform count-based models in sentiment analysis, information retrieval, and semantic texture similarity tasks [17,18]. Although these models are efficient for many NLP tasks, they are inappropriate for domain adaptation problems because documents from different domains have dissimilar word distributions.

Bollegala et al. [5] proposed a distributed representation model of words for the domain adaptation task; words in this model are divided into pivots and non-pivots, and word embeddings for the source and target domains are trained separately. They used hinge loss instead of conditional probability (2) and considered only the relations of pivot and non-pivot words, thereby maximizing the prediction accuracy of non-pivot words in the fixed-length context of a pivot word in each domain by using the following equation:

$$\sum_{C\in\{\mathscr{S},\mathscr{T}\}}\sum_{d\in\mathscr{D}_C}\sum_{(w_p,w_n)\in d}\sum_{w_*\sim p_C(w)} \max\left(0, 1 - v^C_{w_p}\cdot v^C_{w_n} + v^C_{w_p}\cdot v^C_{w_*}\right) \tag{3}$$

where v_w^C is the word embedding of w for domain C of source $\mathscr{S}$ or target $\mathscr{T}$, and w_* is sampled from the 3/4th-powered marginal distribution of non-pivot words in domain C in [5]. The aforementioned objective is regularized by minimizing the differences in pivot word embeddings in the source and target domains. Our method is similar to this method in that it learns distributed representations for domain adaptation. However, our method simultaneously learns document and word representations and does not distinguish between source and target embeddings for words.

In summary, domain adaptation for NLP requires effective embeddings of words and documents. Researchers have developed models learning common features, but these models start with numerical representations rather than textual input. Although the distributed representation framework has been effectively applied to learn word and document embeddings from textual input and has outperformed dictionary-based models, the dictionary-based model is generally used for extracting common features. Therefore, we intend to develop a distributed representation method that indirectly reduces the difference between source and target embedding distributions. Detailed examples and explanations are provided in the following section.

3. Proposed method

Most document representation models suffer from the domain separation problem in which the supports of document embeddings in the source and target domains do not coincide when document representations from different domains are trained simultaneously. For example, this problem can occur in dictionary-based models because the source and target domains share only some of the words. If we use only common words, then document representations can lose a considerable amount of information. Although the Doc2Vec model can learn document embeddings that reflect the relation between words, this model cannot prevent document embeddings from having different distributions across domains in the embedding space. In this study, we focus on developing a document representation model based on the Doc2Vec model to address the domain separation problem.

Distributed representation models can remarkably reduce computational complexity and yield effective representations by training using negative sampling [21]. The negative sampling method is inspired by the NCE method for the efficient learning of word embeddings [22]. We let p_{data} be the training data distribution and p_n be the noise distribution. To apply NCE, a new binary class variable C should be introduced for an auxiliary problem that distinguishes between real and noise data [12], where a new model over an input word w_i, an output word w_j, and C can be specified as follows:

$$p_{joint}(w_j|w_i, C=1) = p_\theta(w_j|w_i), \tag{4}$$

$$p_{joint}(w_j|w_i, C=0) = p_n(w_j). \tag{5}$$

Similar distributions are constructed for the training data, where $p_{train}(w_j|w_i, C=1) = p_{data}(w_j|w_i)$ and $p_{train}(w_j|w_i, C=0) = p_n(w_j)$. We suppose that the negative examples from the noise distributions are k times more frequent than those in the real data ($p_{joint}(C=1) = \frac{1}{k+1}$ and $p_{joint}(C=0) = \frac{k}{k+1}$), and the input instances are independent of the class variable D. Then, the following logistic model can be constructed:

$$\begin{aligned} p_{joint}(C=1|w_i, w_j; \theta) &= \frac{p_\theta(w_j|w_i)}{p_\theta(w_j|w_i) + kp_n(w_j)} \\ &= \sigma(\log p_\theta(w_j|w_i) - \log(kp_n(w_j))) = \sigma(\Delta s_\theta(w_j, w_i)), \end{aligned} \tag{6}$$

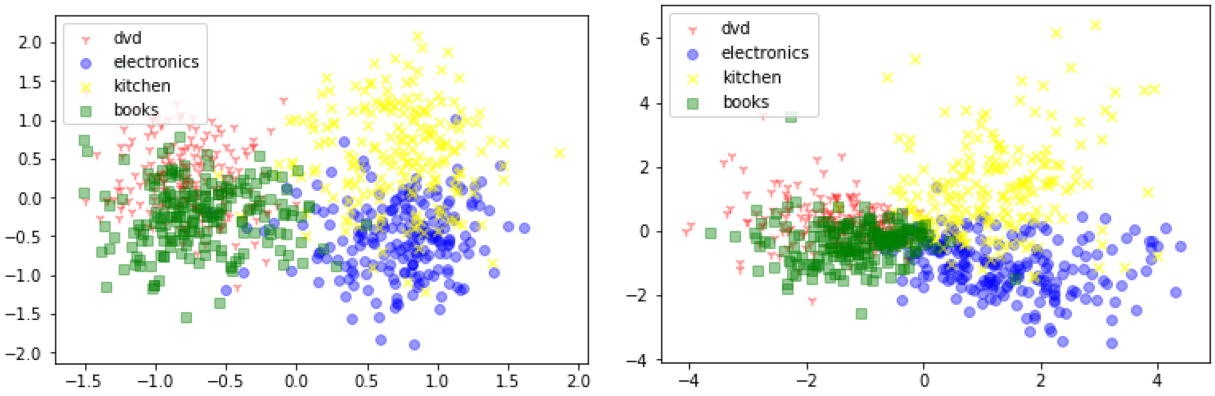


Fig. 2. Document embeddings learned by DBoW and DM from four domains in Amazon review datasets: Book, DVD, Electronics and Kitchen. We randomly selected 100 samples in each domain for clear visualization.

where $\Delta s_\theta(w_j, w_i) = s_\theta(w_j, w_i) - \log \sum_k \exp s(w_k, w_i) - \log(kp_n(w_j))$ that corrects [22] and s_θ is given similar to that in Eq. (1). The model can be fitted by maximizing the log-posterior probability $\log p^i_w(C|w_j)$ averaged over data and noise distribution as follows:

$$\mathbb{E}_{p_{data}(w_i)}\mathbb{E}_{p_{data}(w_j|w_i)}\left[\log \sigma(\Delta s_\theta(w_j, w_i))\right] + k\mathbb{E}_{p_{data}(w_i)}\mathbb{E}_{p_n(w_j)}\left[\log\left(1 - \sigma(\Delta s_\theta(w_j, w_i))\right)\right]. \quad (7)$$

However, Eq. (7) requires intensive computations of the evaluation of the noise distribution p_n for an arbitrary point to calculate the objective function and its gradient. Mikolov et al. proposed the negative sampling method that simplifies NCE by eliminating the evaluation of noise distributions $\Delta s_\theta(w_j, w_i) = \tilde{v}_j^\top v_i$ while maintaining their quality in [21]. Therefore, they maximized Eq. (2) with the following conditional probability:

$$\log p(w_{t+j}|w_t) = \log \sigma\left(\tilde{v}^\top_{w_{t+j}} v_{w_t}\right) - \sum_{i=1}^{k} \mathbb{E}_{w_i \sim p_n(w)}\left[\log \sigma\left(-\tilde{v}^\top_{w_i} v_{w_t}\right)\right] \quad (8)$$

This technique was also extended to train the distributed representations of documents in [18]. The DBoW model has the following objective:

$$\sum_{d\in\mathscr{D}}\sum_{w\in d} \log \sigma\left(\tilde{v}_w^\top v_d\right) + k\sum_{d\in\mathscr{D}} \mathbb{E}_{w'\sim p_n(w')}\left[\log\left(1 - \sigma\left(\tilde{v}^\top_{w'} v_d\right)\right)\right], \quad (9)$$

where $\mathscr{D}$ is the training corpus. The expectation over data distribution is replaced by the training data. As indicated in [15], we can obtain good optimum quality by selecting noise distribution that is similar to the data distribution in certain aspects. This fact was confirmed empirically in [21] because setting the 3/4th powered unigram distribution as the noise distribution produces good results. However, training a model with objective Eq. (9) from documents of multiple domains can separate document embeddings by domain because the model is learned to discriminate between model distribution and the common unigram-based noise distribution. Figure 2 illustrates document embeddings learned by the DBoW and DM models from four domains in Amazon review datasets. As shown in the figure, the document embeddings of each domain are clustered rather than evenly spread in both distributed representation models. This result can be explained from Eq. (9). Negative sampling method moves a document embedding close to the embeddings of the consisting words and farther from the embeddings of words from the noise distribution. Given that the unigram-based noise distribution is used, the document embeddings are less affected by the words that occurred frequently in all domains; the effect of stop words is already

reduced by subsampling [20]. Therefore, the document embeddings from different domains are separated from the common distribution. This property of embeddings is inappropriate for domain adaptation tasks because of the high $\mathscr{H}$-divergence [1].

To solve this problem, we introduce a new auxiliary variable D for the domains, where $D = \mathscr{S}$ denotes the source domain, whereas $D = \mathscr{T}$ denotes the target domain. We assume that the domain variable D is independent of the class variable C. Then, we can construct a new model as follows:

$$p(w|d, C = 1, D) = p(w|d, C = 1) = p_{data}(w|d), \tag{10}$$

$$p(w|d, C = 0, D) = p(w|C = 0, D) = p_n(w|D). \tag{11}$$

The second equality in Eq. (10) holds because the domain variable D is automatically determined given a document d. However, noise distribution is dependent on the domain variable D because this distribution is independent of the input variable d as shown in Eq. (5). The joint noise distribution $p_n(w, d)$ can be induced as the following equations.

$$\begin{aligned}
p(w, d|C = 0) &= p(w, d, D = \mathscr{S}|C = 0) + p(w, d, D = \mathscr{T}|C = 0) \\
&= p(w|d, D = \mathscr{S}, C = 0)p(d|D = \mathscr{S})p(D = \mathscr{S}) \\
&\quad + p(w|d, D = \mathscr{T}, C = 0)p(d|D = \mathscr{T})p(D = \mathscr{T}) \\
&= p_n(w|D = \mathscr{S})p(d|D = \mathscr{S})p(D = \mathscr{S}) \\
&\quad + p_n(w|D = \mathscr{T})p(d|D = \mathscr{T})p(D = \mathscr{T})
\end{aligned}$$

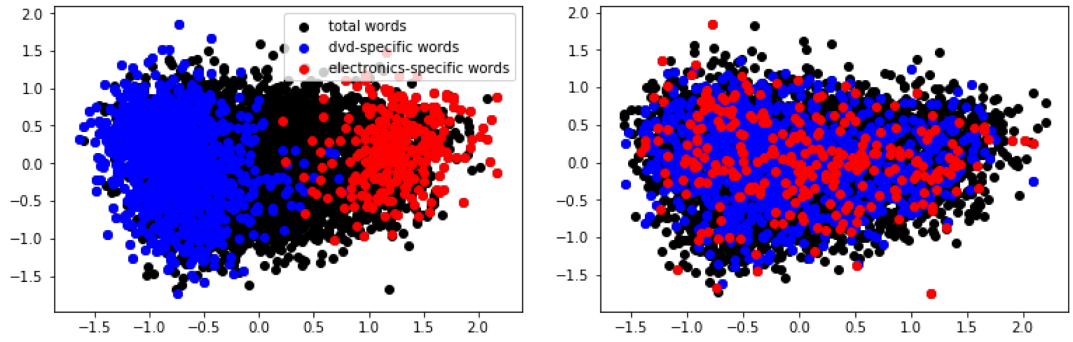
We can construct the following objective of training document embeddings for domain adaptation by replacing the part of the train distributions ($p_{data}(w, d)$ and $p_n(w, d)$) with the samples:

$$\sum_{d \in \mathscr{D}_\mathscr{S} \vee \mathscr{D}_\mathscr{T}} \sum_{w \in d} \log \sigma\left(\tilde{v}_w^\top v_d\right) + k \sum_{D \in \{\mathscr{S}, \mathscr{T}\}} \sum_{d \in \mathscr{D}_D} \mathbb{E}_{w' \sim p_n(w'|D)} \left[\log\left(1 - \sigma\left(\tilde{v}_{w'}^\top v_d\right)\right)\right], \tag{12}$$

where the 3/4th-powered unigram distribution of domain D is used for $p_n(w'|D)$. We alternatively maximize our objective Eq. (12) and the skip-gram objective Eq. (8) by using the stochastic gradient descent method [27]. In Eq. (12), the domain-dependent noise distribution $p_n(w|D)$ can improve embedding quality by providing a more similar distribution to the data distribution $p_{data}(w|d)$ than the marginal noise distribution $p_n(w)$. Moreover, the document embeddings can be closer to the important words rather than farther from the words that frequently appear in both domains. Addressing the domain separation problem of document embeddings is important. Algorithm 1 shows the Pseudo code of our approach.

Algorithm 1 Pseudocode of learning the indiscriminate document embeddings

1: **Input:** Document datasets $\mathscr{D}_\mathscr{S} = \{D_1^\mathscr{S}, \ldots, D_N^\mathscr{S}\}$ and $\mathscr{D}_\mathscr{T} = \{D_1^\mathscr{T}, \ldots, D_N^\mathscr{T}\}$ from source ($\mathscr{S}$) and target ($\mathscr{T}$) domains, minimum word count (min_count)
2: **Output:** Indiscriminate document embeddings $\{v_d : d \in \mathscr{D}_\mathscr{S} \vee \mathscr{D}_\mathscr{T}\}$ and word embeddings $\{v_w : w \in \mathscr{W}\}$
3: Construct word dictionary $\mathscr{W} = \{w : \textit{count}(w) \geqslant \textit{min\_count}, w \in D_j^i, i \in \{\mathscr{S}, \mathscr{T}\}, j = 1, \ldots, N\}$
4: Obtain the cumulative frequency tables $F_\mathscr{S}$ and $F_\mathscr{T}$ sorted by frequency
5: Initialize document embeddings $\{v_d\}$ and word embeddings $\{v_w\}$
6: **procedure** TRAINEMBEDDS($\mathscr{D}_\mathscr{S}, \mathscr{D}_\mathscr{T}, \{v_d\}, \{v_w\}, F_\mathscr{S}, F_\mathscr{T}$)
7: Set the noise distribution $p_n(w|D)$ as the 3/4th unigram distribution of $F_D, D \in \{\mathscr{S}, \mathscr{T}\}$
8: Maximize the objective (12)
9: **end procedure**



(a) Domain-independent noise distribution (b) Domain-dependent noise distribution

Fig. 3. Word embeddings from two noise distributions. Black dots represent common words, whereas black and blue dots represent domain-specific words.

In case of word embeddings, the domain-dependent noise distribution can help identify the meaningful relations between words because domain information is crucial for determining the importance of a word. Therefore, we inspect the effect of the domain-dependent noise distribution on word embeddings through the following simple example. We learn document and word embeddings for two cases depending on whether the skip-gram model also introduces the domain variable D. In this example, domain-specific words are those that appear only in one of the domains. We inspect the embeddings of domain-specific words to show the property of the trained word distributions. We compare word embeddings from domain-dependent and domain-independent noise distributions by visualizing word embeddings to determine the influence of the change in the noise distribution in Eq. (8) on the distribution of the embeddings of domain-specific words. We train the representations on the basis of the experimental design presented in Section 4.2. Figure 3 shows the word embeddings of DVD and Electronics reviews in the Amazon dataset.

As expected, the embeddings of domain-specific words in Fig. 3a are separated depending on domains, whereas those in Fig. 3b are not. When the domain-dependent noise distribution in Eq. (8) is applied, the embeddings of all the words become inseparable. Meanwhile, the embeddings of domain-specific words become separable even if the embeddings of the other words are shared. This tendency is consistently observed in other pairs, the results of which are presented in Appendix B. The resulting document embeddings can possess different attributes because they are affected by the word embeddings. In the following section, we verify the effectiveness of the document embeddings obtained using the proposed method through several experiments on real data.

4. Experiments

We evaluated our approach through the widely used Amazon dataset, which contains customer reviews with grades for purchased products. We learned document representations and performed visualization and classification tasks. We expect that the appropriate document representations from different domains would be overlapped and effective for a specific task (sentiment classification in our case). Document representations would be evaluated from this perspective through visualization and classification. We focused on learning the common features of source and target domains with similar distribution. Thus, we postulated the use of the same classifier. Accordingly, we trained the classifier only with source data and tested it with target data.

Table 1
Number of total words and number of domain-specific words in every pair (we only counted words that appear more than 10 times in the documents)

Data sets A & B	Total words	Domain-specific words of A	Domain-specific words of B
Book & DVD	5784	59	62
Book & Electronics	4682	651	184
DVD & Electronics	4797	758	196
Kitchen & Electronics	3435	218	223
Kitchen & Book	4675	170	760
Kitchen & DVD	4777	183	850

4.1. Data description

In this experiment, we used four categories of the Amazon review dataset [3], namely, Book, DVD, Kitchen and Electronics. We produced 6 datasets for domain adaptation, which consisted of 6,000 documents from 2 categories: (Book, DVD), (Book, Electronics), (DVD, Electronics), (Kitchen, DVD), (Kitchen, Electronics), and (Kitchen, Book). For the classification task, we transformed the score between 1 and 5 to a sentiment label, where we regarded 1–2 points as negative sentiments and 4–5 points as positive sentiments. Moreover, our domain adaptation datasets were balanced for sentiment labels and domains.

Table 1 shows the numbers of total words and domain-specific words for all pairs. As shown in the table, Book and DVD have various words, whereas Kitchen and Electronics have less domain-specific words.[1] Consequently, Book and DVD share many common words, and Kitchen and Electronics have a relatively high proportion of common words (87.16%).

4.2. Experimental design

Most document representation learning methods for domain adaptation are task dependent because they use the source labels and learn new document representations from the numerical representations of documents and not from the textual inputs. By contrast, the proposed method is purely unsupervised (task independent) and learns representations from scratch (from the textual inputs). Therefore, we first compared the distribution of document embeddings from two domains. We also performed sentiment classification tasks to verify that the learned document embeddings (d) effectively capture the useful information from the original textual inputs and have similar conditional distributions $p(y|d, D = \mathscr{S}) \approx p(y|d, D = \mathscr{T})$ for sentiment label y.

We compared our proposed method with different document representation methods, namely, BoW, TF-IDF, DBoW, DM, and SCL. BoW and TF-IDF are dictionary-based models; hence, we selected the most frequent 5,000 vocabularies as the dictionary. We reduced the dimension of document representations from dictionary-based models to 200 by using principal component analysis (PCA) to address the curse of dimensionality. Different from dictionary-based models, our methods and the distributed models (DBoW and DM) are required to set several hyper-parameters to learn document representations. We determined hyper-parameters by training DBoW model and cross-validating the learned representations

[1] In this experiment, we considered only the words that appear more than 10 times in the dataset without stemming. The number of domain-specific words counts the words that appear only in a specific category. For example, in the dataset of Book and DVD, 5784 words appear more than 10 times, 59 words appear only in Book reviews, and 62 words appear only in DVD reviews.

with support vector machine (SVM) classifier. The hyper-parameter settings we explored were (window = {3, 5}, minimum count = {5, 10, 20}, negative count = {5, 10}, and dimension = {100, 200}). We illustrate all these results in Appendix A. We selected the good (high cross-validation accuracy) and stable (consistently good over domains) classification performance of hyper-parameter pair (window = 3, minimum count = 5, negative count = 5, and dimension = 200) in our models and the comparative distributed models. We also compared our model with a popular domain adaptation model for document representations, namely, SCL [4]. SCL identifies correspondences among features from different domains by modeling their correlations with pivot features that behave in the same way for sentimental analysis in both domains. For our proposed approach, we trained the embeddings of words and documents with domain-dependent noise distribution. Therefore, we obtained the experimental results from six methods.

Representation learning was conducted with six combinations of four domains in the models except SCL[2] because the role of the source and target datasets is not separated when learning representations in a purely unsupervised setting. Subsequently, we visualized the document embeddings from the models and quantitatively measured the differences in their distributions after reducing the dimension to two via PCA to determine whether the source and target representations are combined effectively.

We used proxy A-distance (PAD), which measures the difference between the data distributions and the domains. The A-distance is a dissimilarity measure among different distributions that was suggested in [2]. PAD is an estimate of H-divergence, which is immediately computable from the error as follows:

$$\hat{d}_{\mathscr{H}}(\mathfrak{U}, \mathfrak{U}') = 2\left(1 - \min_{h \in \mathscr{H}} \left[\frac{1}{m} \sum_{\mathbf{x}: h(\mathbf{x})=0} I[x \in \mathfrak{U}] + \frac{1}{m} \sum_{\mathbf{x}: h(\mathbf{x})=1} I[\mathbf{x} \in \mathfrak{U}']\right]\right), \tag{13}$$

where $\mathscr{H}$ is a symmetric hypothesis class of binary classifiers, samples $\mathfrak{U}$, $\mathfrak{U}'$ of size m, and $I[x \in \mathfrak{U}]$ is the binary indicator variable. This divergence can be obtained by finding minimum error for the binary classification problem of distinguishing source from target instances. However, exploring the entire hypothesis space $\mathscr{H}$ is intractable. Therefore, we approximated this measure by restricting the hypothesis class. We practically measured the classification error ϵ of the support vector machine (SVM) classifier that was trained to discriminate between points sampled from different domains, where PAD is defined as $d_{PAD} = 2(1 - 2\epsilon)$. PAD ranges from -1 to 1, but it is usually greater than 0 because most binary classifiers have test errors that are lower than 0.5. Small PAD indicates that the difference between two distributions is small. After randomly splitting the training and test data with a test ratio of 0.2, we trained the SVM classifier with the training data and calculated PAD with the remaining test data, where a linear kernel was used and the soft margin parameter was fixed to 10 as indicated in [10]. We measured the dissimilarity between source and target distributions by using PAD because PAD requires only data samples from source and target distributions, whereas other dissimilarity measures require prior estimation of the distributions.

We applied document representations to sentimental prediction to determine if the representations can be transferred to sentimental labels. We conducted 12 crossdomain adaptation experiments with 4 domains. We used the rbf SVM classifier with 0.01 as kernel parameter and 10 as margin parameter for classification, where the source data were used for training. Then, we tested the model directly on the target data. We compared classification accuracies in the case of 2D and 200D representations. Finally,

[2]We excluded the SCL method in this unsupervised experiment because it disentangled the source and target domains and used the source labels when extracting the pivot features.

Table 2
Proxy A-distance of 2-dimensional data

Source & target	BoW	TFIDF	DM	DBoW	Proposed method
B & D	**0.0364**	1.7480	1.5440	1.5560	0.2480
B & E	0.5640	1.8396	1.8636	1.9116	**0.2360**
D & E	0.6360	1.7280	1.8476	1.8676	**0.0200**
K & E	**−0.0324**	0.1720	1.5480	1.5720	0.0520
K & B	0.5480	1.8076	1.8556	1.9456	**0.5440**
K & D	0.5560	1.6160	1.5480	1.8956	**0.0520**

we trained the new common embeddings and the sentimental classifiers by applying a deep learning-based domain adaptation model to the obtained document representations. We used the DANN model because it is simple but comparable to other state-of-the-art algorithms [10]. We also examined the convergence of the target test error and the classification accuracies to train the DANN model. Table 2: Proxy A-distance of 2D data.

4.3. Results

4.3.1. Visualization

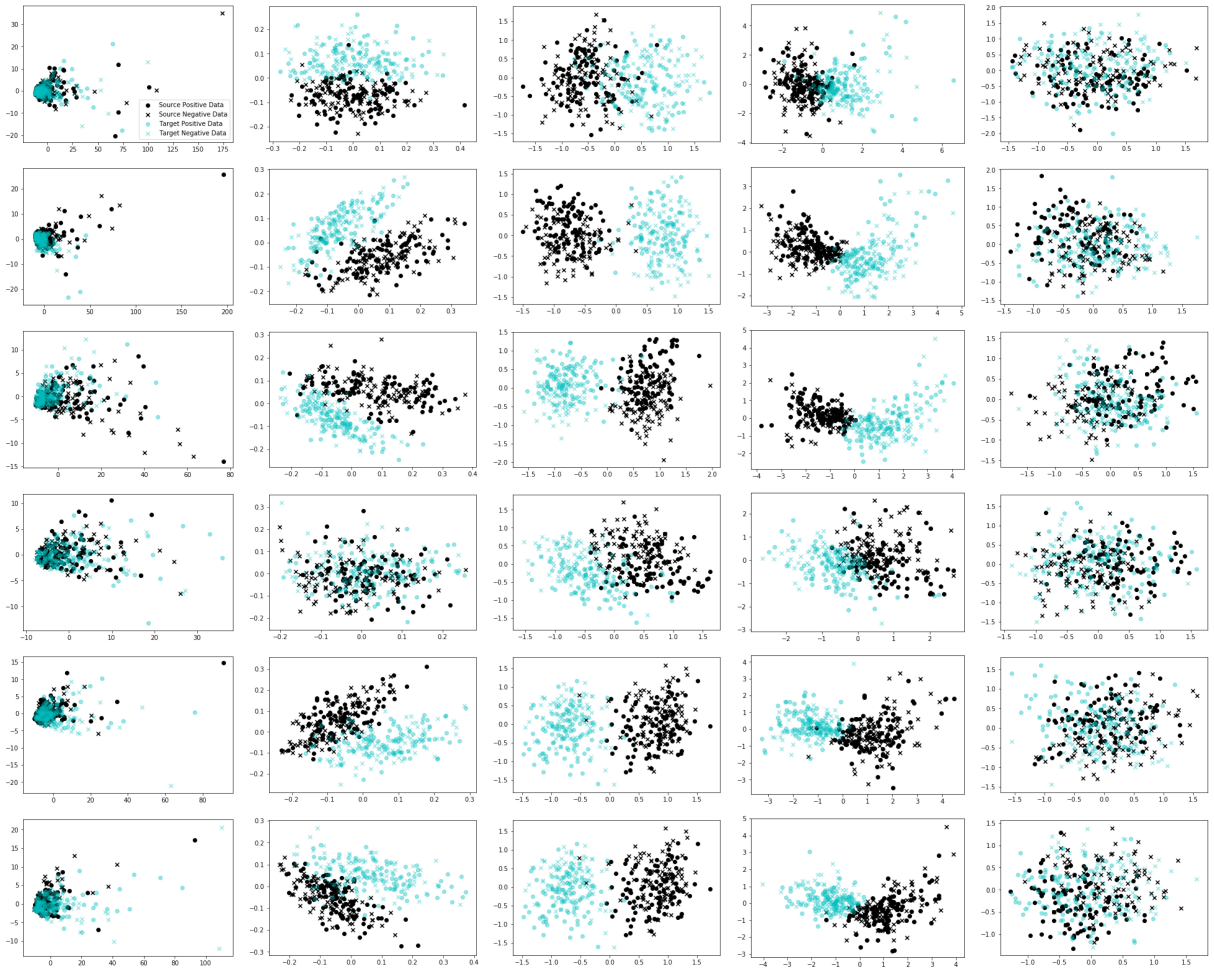
We visualized document representations from five models after applying PCA. Figure 4 shows the visualization results. Each row represents a pair of two domains (A and B). The black circle corresponds to a positive A document, the black cross corresponds to a negative A document, and the blue color is for the B domain. The support regions of the domains in the Book and Electronics pair and the DVD and Electronics pairs are separated in the visualization results of BoW. The representations of TF-IDF better separate each domain's representation than those of BoW. Distributed representation-based methods such as DBoW and DM show similar results that the document embeddings are divided according to the domains. The proposed method is consistently good at mixing the document representations from two domains in all the cases, whereas the representations of DBoW are separated according to the domains despite having similar objective functions except for the noise distribution of the output words.

We used PAD to quantitatively demonstrate that the proposed method results in indistinguishable distributions of document embeddings with respect to the domains. The distributions of the source and target representations are similar when PAD is low based on the definition. The PADs of 2D representations are provided in Table 2.

Table 2 shows that the proposed method consistently exhibits low PAD measures regardless of the dataset pairs. As mentioned in Section 4.1, BoW can have low PADs on the Book and DVD and Kitchen and Electronics pairs because these pairs have relatively high proportion of common words. The DBoW and TF-IDF methods have high PADs for all the dataset pairs, thereby indicating that these methods can separate document distributions to consider word distribution based on the total document corpus. From these visualizations and the PAD results, we can conclude that our novel domain-dependent noise distribution of words and documents is effective for learning the document embeddings with similar distributions across domains.

4.3.2. Sentiment classification

Domain adaptation requires document representations that overlap across domains in terms of unsupervised learning regardless of the specific NLP task. However, these representations should also conceive the effective information about texts. Thus, we evaluated document representations through a sentiment classification task. Experiments were conducted on the 2D and 200D representations presented in Section 4.3.1. We measured the performance of the SVM classifier, which was trained on the



(a) BoW (b) TF-IDF (c) DBoW (d) DM (e) Proposed method

Fig. 4. 2D plots of the document representations of the Amazon dataset. Each row contains a pair of two categories (A and B) in the following order: Book and DVD, Book and Electronics, DVD and Electronics, Kitchen and Electronics, Kitchen and Book, and Kitchen and DVD. We randomly selected 100 samples in each domain for clear visualization.

source data and tested on the target data for 12 source and target pairs, to determine if the representation methods had learned the document embeddings that were informative to the sentiment analysis and had the invariant classifier across domains without applying any covariate shift techniques to the classifier. Table 3 presents the domain adaptation accuracies of the Amazon review datasets in 2D data.

Table 3 shows that the proposed method consistently exhibits high accuracies in all the experimental pairs compared with other word representations. We can infer that BoW, TF-IDF, DM, and SCL models have difficulty in discriminating among the sentiments of document embeddings because their accuracies are nearly 50% in binary classification. The DBoW representation shows acceptable results in a few pairs but does not perform consistently in all the pairs. From the aforementioned results, we can conclude that our suggested methods combine source and target distributions, thereby maintaining their sentimental information. Accordingly, we can determine that the proposed method extracts sentiment transferable features among different domains.

Table 3
Domain adaptation accuracy of Amazon review datasets in 2D data

Source → target	BoW	TF-IDF	DM	DBoW	SCL	Proposed method
B → D	51.95%	52.80%	50.80%	52.52%	53.95%	**61.04%**
B → E	52.12%	56.47%	49.84%	51.60%	58.05%	**61.19%**
B → K	50.56%	56.47%	50.16%	54.07%	59.03%	**63.95%**
D → B	50.31%	50.96%	52.40%	54.44%	57.30%	**61.19%**
D → E	53.23%	54.60%	49.39%	60.16%	57.65%	**62.96%**
D → K	52.48%	55.27%	51.31%	62.80%	60.05%	**67.20%**
E → B	51.59%	54.64%	52.52%	53.83%	55.30%	**69.64%**
E → D	51.55%	53.15%	49.91%	71.79%	53.95%	**70.44%**
E → K	53.55%	56.20%	61.39%	71.39%	60.65%	**71.40%**
K → B	50.60%	54.52%	50.80%	68.27%	69.05%	**75.20%**
K → D	49.91%	53.35%	51.55%	**72.16%**	55.55%	69.27%
K → E	52.43%	56.04%	60.24%	67.52%	57.60%	**71.16%**

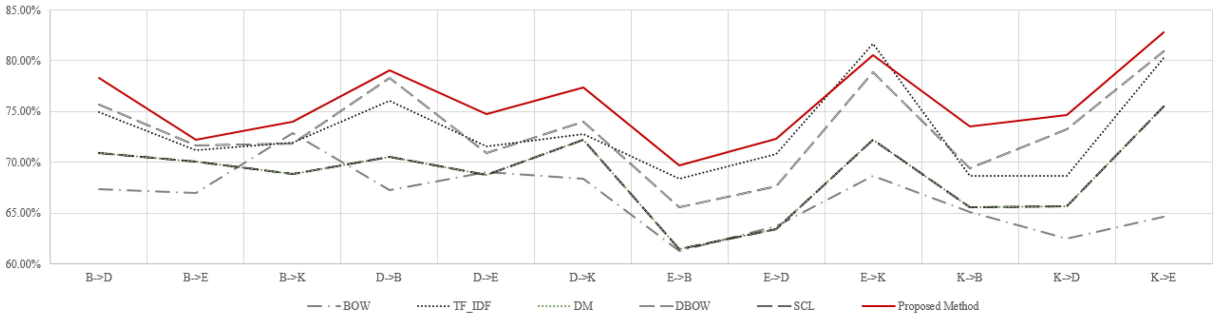


Fig. 5. Sentiment classification results of document representations.

We also performed domain adaptation sentimental analysis by using the original representations of the document representation model. We used PCA to document the embeddings of the BoW and TF-IDF models for reducing the dimension to 200 to obtain the same dimension as distributed representations because the dimensions of the original representations are too high (5000). We reduced the dimension of SCL to 50 dimensions because it shows best result in these dimensions. The results of the sentimental analysis of the 200D document representations are illustrated in Fig. 5.

Figure 5 shows that our suggested models outperform other representations in all the dataset pairs. In particular, the proposed method achieves better results than the other methods. The Doc2vec models exhibit similar patterns, but the DBoW models demonstrate slightly better performance than DM. TF-IDF also obtains similar results, whereas the BoW models present the worst performance in eight experiments. Many common words exist between the Book and DVD, as well as Kitchen and Electronics pairs. All the results show high performance when the word distributions in the source and target domains are similar, such as (Book → DVD), (Electronics → Kitchen), and (Kitchen → Electronics), and our suggested method does not demonstrate considerable performance improvement compared with the other methods. By contrast, our methods considerably outperforms the other methods when the word distributions differ, such as (DVD → Electronics), (DVD → Kitchen), and (Kitchen → Book).

4.3.3. DANN application

As previously mentioned, the document embeddings from the proposed methods can be applied to other domain adaptation methods starting from the numerical vectors because our methods can learn numerical representations from textual input. To verify this useful feature, we also trained the suggested

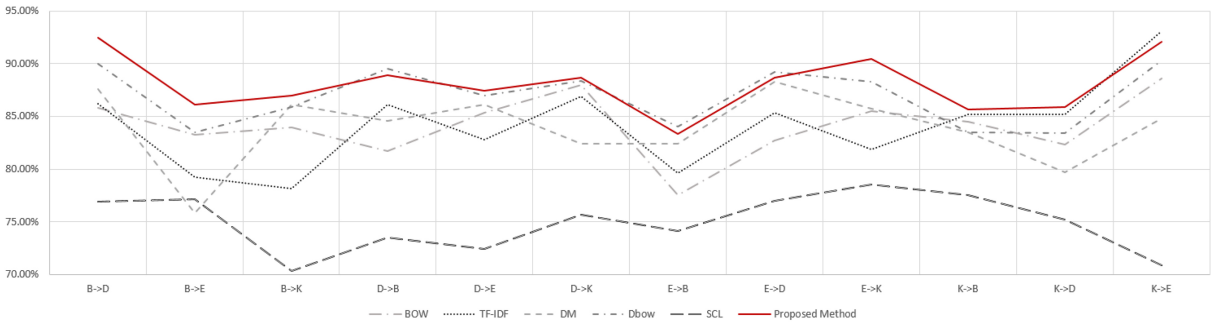
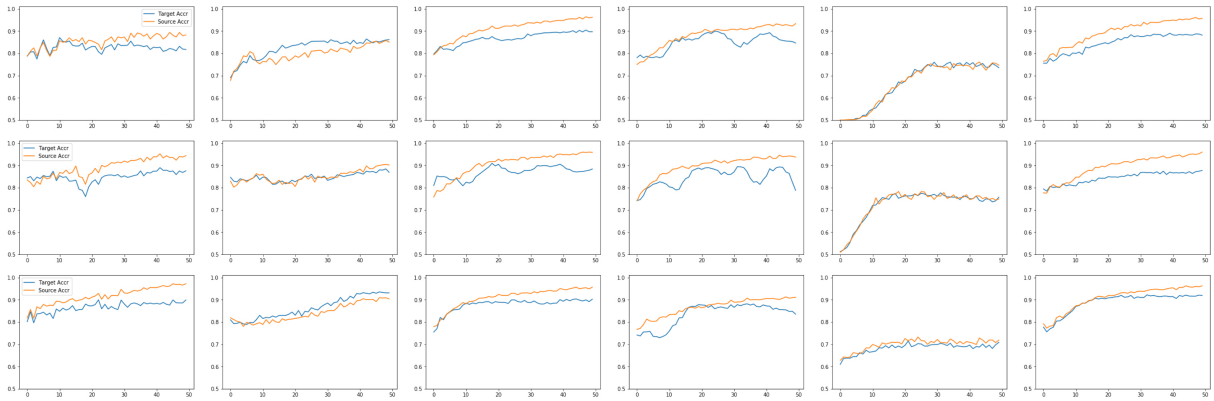


Fig. 6. Sentiment classification results of document representations by using DANN model.



(a) BoW (b) TF-IDF (c) DBoW (d) DM (e) SCL (f) Proposed method

Fig. 7. Accuracy by epoch on document representations of Amazon dataset in experiments (DVD → Book), (DVD → Kitchen), and (Kitchen → Electronics). Orange and blue lines refer to source and target training accuracies, respectively.

new common embeddings and sentimental classifier by applying the DANN model. We aimed to ascertain whether the suggested text representations could improve the accuracy of the sentimental classifier when applied to the latest deep learning-based domain adaptation model. In the DANN model, we used two feature extraction layers (input dimension → 200 → 100), which had fully connected linear layers with the ReLU activation function. In addition, we set two layers (100 → 100 → 1) with the sigmoid function for the domain and sentimental classifiers. We stopped training the model when the accuracy of the source data approached 1.

Figure 6 shows the results of the DANN application. Notably, the proposed methods have high performance in all the pairs. However, DBoW and TF-IDF achieve slightly better results than the proposed methods in some cases, such as (DVD → Book), (DVD → Kitchen), (Electronics → DVD), and (Kitchen → Electronics). One possible reason for this finding is that the DANN model attempts to obtain new common embeddings that are indiscriminate with respect to the source and target domains to offset the advantageous properties of our representations. Nevertheless, the proposed method performs the best for the eight experimental pairs and consistently exhibits good performance.

We also visualized accuracy by epoch in Fig. 7 to examine the convergence aspect of adversarial training. We showed only the cases in which the proposed method demonstrates inferior performance, that is, (DVD → Book), (DVD → Kitchen), and (Kitchen → Electronics). The visualization of accuracy by

epoch for other dataset pairs is presented in Appendix C. Figure 7 shows that our model exhibits a consistent increasing trend compared with the other methods in source and target training accuracies. BoW and TF-IDF show fluctuating accuracy per epoch despite the decaying learning rate in training. In the DBoW and DM models, the accuracy of the source data consistently increases, whereas the accuracy of the target data unstably changes. SCL models also show consistent improvement but with low accuracy rates. Although our methods exhibit slightly low performance in these datasets, their target accuracy follows the source accuracy and stably increases. The target training error of the proposed methods is insensitive to the feature extraction phase in DANN because our document embeddings of the source and target data are already similar. However, the target accuracy of the other methods depends considerably on the feature extraction of DANN. Therefore, the corresponding graph fluctuates through epochs. We can conclude that, although the performance of our proposed model is not the best in all the experimental pairs, our methods not only demonstrate robust performance but also help in the stable convergence of adversarial training.

5. Conclusion

In this study, we proposed a novel distributed representation learning method of words and documents for the domain adaptation by exploring the negative sampling method and utilizing useful properties from the method. Our model can obtain document embeddings with similar distributions for different domains in a purely unsupervised manner from textual input by using sophisticated noise distribution. We showed that the proposed method combined source and target embeddings by visualizing them in 2D and calculating PAD measures. We conducted a sentiment classification task in which the document embeddings of the source domain were used to train the SVM classifier and then the trained classifier was directly applied to target embeddings. The proposed models outperformed other comparative methods in 2D and 200D embeddings. We showed that our method can be efficiently applied to one of the latest deep learning algorithms, that is, DANN. We expect that the proposed method can be extended to domain adaptation methods for various NLP tasks, such as when documents are obtained from different domains.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (MEST) (No. 2016R1A2B3014030, No. 2017R1A5A1015626 and No. 2018R1D1A1A02085851).

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira and J.W. Vaughan, A theory of learning from different domains, *Machine Learning* **79**(1) (2010), 151–175.

[2] S. Ben-David, J. Blitzer, K. Crammer and F. Pereira, Analysis of representations for domain adaptation, in: *Advances in Neural Information Processing Systems*, 2007, pp. 137–144.

[3] J. Blitzer, M. Dredze, F. Pereira et al., Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, in: *ACL*, Vol. 7, 2007, pp. 440–447.

[4] J. Blitzer, R. McDonald and F. Pereira, Domain adaptation with structural correspondence learning, in: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2006, pp. 120–128.

[5] D. Bollegala, T. Maehara and K.-I. Kawarabayashi, Unsupervised cross-domain word representation learning, arXiv preprint arXiv:1505.07184, 2015.
[6] D. Bollegala, T. Mu and J.Y. Goulermas, Cross-domain sentiment classification using sentiment sensitive embeddings, *IEEE Transactions on Knowledge and Data Engineering* **28**(2) (2016), 398–410.
[7] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan and D. Erhan, Domain separation networks, in: *Advances in Neural Information Processing Systems*, 2016, pp. 343–351.
[8] M. Chen, Z. Xu, K. Weinberger and F. Sha, Marginalized denoising autoencoders for domain adaptation, arXiv preprint arXiv:1206.4683, 2012.
[9] H. Daumé III, A. Kumar and A. Saha, Frustratingly easy semi-supervised domain adaptation, in: *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, Association for Computational Linguistics, 2010, pp. 53–59.
[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand and V. Lempitsky, Domain-adversarial training of neural networks, *Journal of Machine Learning Research* **17**(59) (2016), 1–35.
[11] X. Glorot, A. Bordes and Y. Bengio, Domain adaptation for large-scale sentiment classification: A deep learning approach, in: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 513–520.
[12] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, MIT press, 2016.
[13] R. Gopalan, R. Li and R. Chellappa, Unsupervised adaptation across domain shifts by generating intermediate data representations, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(11) (2014), 2288–2302.
[14] A. Gretton, A.J. Smola, J. Huang, M. Schmittfull, K.M. Borgwardt and B. Schölkopf, Covariate shift by kernel mean matching, MIT press, 2009.
[15] M. Gutmann and A. Hyvärinen, Noise-contrastive estimation: A new estimation principle for unnormalized statistical models, in: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 297–304.
[16] T. Kanamori, S. Hido and M. Sugiyama, Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection, in: *Advances in Neural Information Processing Systems*, 2009, pp. 809–816.
[17] J.H. Lau and T. Baldwin, An empirical evaluation of doc2vec with practical insights into document embedding generation, arXiv preprint arXiv:1607.05368, 2016.
[18] Q. Le and T. Mikolov, Distributed representations of sentences and documents, in: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.
[19] M. Long, J. Wang, Y. Cao, J. Sun and S.Y. Philip, Deep learning of transferable representation for scalable domain adaptation, *IEEE Transactions on Knowledge and Data Engineering* **28**(8) (2016), 2027–2040.
[20] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781, 2013.
[21] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in: *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.
[22] A. Mnih and K. Kavukcuoglu, Learning word embeddings efficiently with noise-contrastive estimation, in: *Advances in Neural Information Processing Systems*, 2013, pp. 2265–2273.
[23] M. Sugiyama, S. Nakajima, H. Kashima, P.V. Buenau and M. Kawanabe, Direct importance estimation with model selection and its application to covariate shift adaptation, in: *Advances in Neural Information Processing Systems*, 2008, pp. 1433–1440.
[24] M. Xiao and Y. Guo, Feature space independent semi-supervised domain adaptation via kernel matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**(1) (2015), 54–66.
[25] T. Yao, Y. Pan, C.-W. Ngo, H. Li and T. Mei, Semi-supervised domain adaptation with subspace learning for visual recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2142–2150.
[26] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger and S. Saminger-Platz, Central moment discrepancy (cmd) for domain-invariant representation learning, arXiv preprint arXiv:1702.08811, 2017.
[27] M. Zinkevich, M. Weimer, L. Li and A.J. Smola, Parallelized stochastic gradient descent, in: *Advances in Neural Information Processing Systems*, 2010, pp. 2595–2603.

Appendix

A. Hyper-parameter selection

We needed to choose and fix the hyper-parameter of Doc2Vec-based suggested method and comparative models. Distributed representation methods have various hyper-parameter settings, and this setting is relevant to the performance of the model.

We must find hyper-parameters for our research that can effectively learn representation of the Amazon Review data for sentimental analysis. In each category, we used DBoW-based representation learning to predict the sentiment of the review. We repeated this experiment 10 times and measured average accuracy and standard deviation for each setting.

Table 4 is the results of the hyper-parameter experiments. In this table, hyper-parameter "window" refers to the maximum distance between the current and predicted word within a sentence. "Minimum count" is a criterion; all words with total frequency lower than this criterion will be ignored by the model. "Negative count" indicates the number of noise words that should be drawn in negative sampling. "Dimension" is the dimensionality of the feature vectors. ** refers to the best value in each category, and * indicates the second best setting. All the results are rounded off to four decimal places.

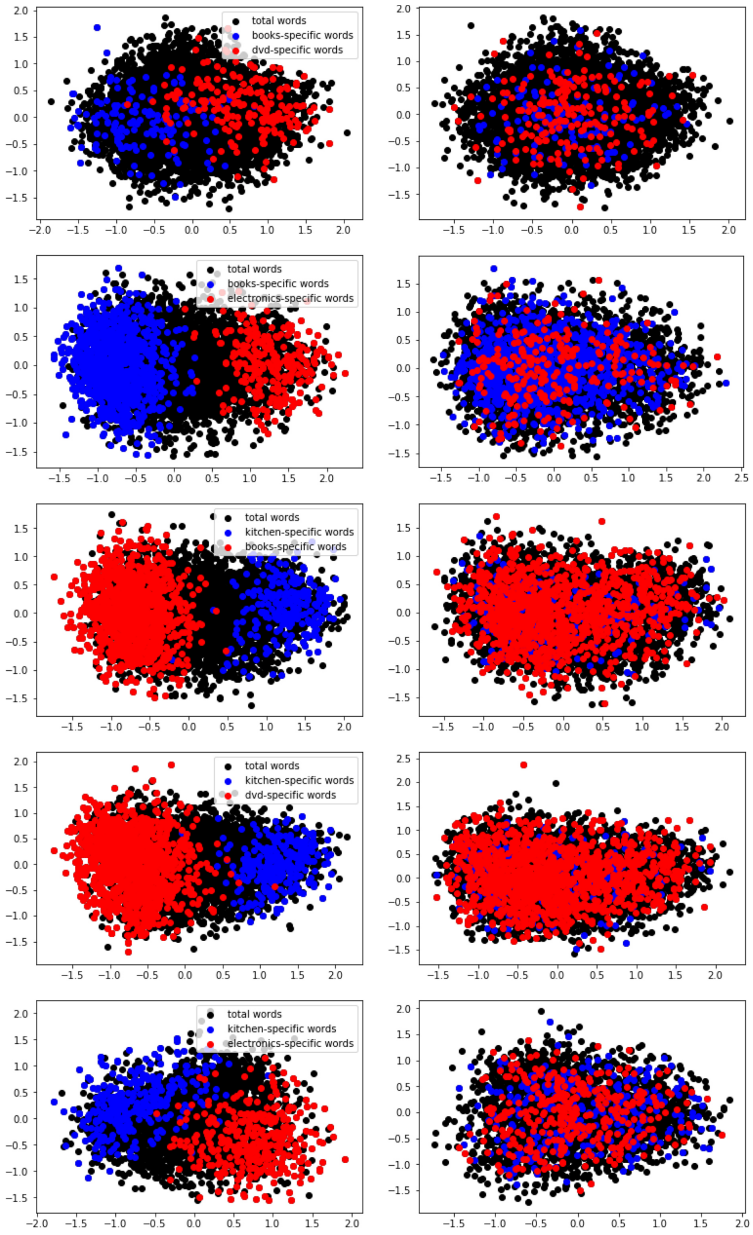
We selected hyper-parameter (window = 3, minimum count = 5, negative count = 5, and dimension = 200) in our research. This setting shows robust performance in all four domains and is the optimal in average.

Table 4
Result of hyper-parameter selection experiments

Hyper-parameter				Books	DVD	Kitchen	Electronics	Average
Window	Min word	Negative	Dimension	Accuracy (std)	Accuracy (std)	Accuracy (std)	Accuracy (std)	Accuracy (std)
3	5	5	100	0.840 (0.010)	0.859 (0.014)	0.885 (0.014)	0.888 (0.010)	0.868 (0.012)
3	5	5	200	0.840* (0.006)	0.865** (0.013)	0.889 (0.014)	0.897* (0.011)	0.873** (0.011)
3	5	10	100	0.840 (0.010)	0.859 (0.010)	0.891** (0.012)	0.893 (0.011)	0.871 (0.011)
3	5	10	200	0.841** (0.008)	0.860 (0.012)	0.888 (0.013)	0.890 (0.012)	0.870 (0.011)
3	10	5	100	0.836 (0.009)	0.858 (0.010)	0.879 (0.013)	0.877 (0.010)	0.862 (0.011)
3	10	5	200	0.832 (0.011)	0.852 (0.014)	0.885 (0.012)	0.880 (0.010)	0.862 (0.012)
3	10	10	100	0.830 (0.011)	0.855 (0.012)	0.871 (0.012)	0.878 (0.011)	0.858 (0.012)
3	10	10	200	0.838 (0.013)	0.859 (0.009)	0.884 (0.011)	0.880 (0.014)	0.865 (0.012)
3	20	5	100	0.813 (0.010)	0.844 (0.015)	0.865 (0.013)	0.871 (0.011)	0.848 (0.012)
3	20	5	200	0.825 (0.011)	0.845 (0.011)	0.868 (0.017)	0.872 (0.009)	0.853 (0.012)
3	20	10	100	0.816 (0.010)	0.844 (0.014)	0.874 (0.014)	0.869 (0.010)	0.851 (0.012)
3	20	10	200	0.823 (0.009)	0.841 (0.012)	0.872 (0.017)	0.871 (0.008)	0.852 (0.011)
5	5	5	100	0.841 (0.011)	0.856 (0.014)	0.890* (0.013)	0.887 (0.015)	0.868 (0.013)
5	5	5	200	0.839 (0.008)	0.864* (0.011)	0.886 (0.013)	0.895 (0.010)	0.871 (0.010)
5	5	10	100	0.835 (0.011)	0.855 (0.011)	0.883 (0.010)	0.886 (0.012)	0.865 (0.011)
5	5	10	200	0.841 (0.012)	0.861 (0.014)	0.888 (0.013)	0.898** (0.011)	0.872* (0.012)
5	10	5	100	0.833 (0.013)	0.848 (0.012)	0.876 (0.011)	0.878 (0.013)	0.859 (0.012)
5	10	5	200	0.831 (0.013)	0.851 (0.010)	0.874 (0.012)	0.881 (0.012)	0.859 (0.012)
5	10	10	100	0.834 (0.013)	0.852 (0.013)	0.880 (0.013)	0.873 (0.013)	0.860 (0.013)
5	10	10	200	0.835 (0.014)	0.851 (0.011)	0.882 (0.014)	0.880 (0.011)	0.862 (0.013)
5	20	5	100	0.820 (0.013)	0.840 (0.016)	0.863 (0.014)	0.864 (0.009)	0.847 (0.013)
5	20	5	200	0.825 (0.011)	0.843 (0.015)	0.872 (0.013)	0.871 (0.012)	0.853 (0.013)
5	20	10	100	0.818 (0.011)	0.835 (0.013)	0.867 (0.013)	0.867 (0.011)	0.847 (0.012)
5	20	10	200	0.819 (0.012)	0.843 (0.012)	0.871 (0.012)	0.872 (0.012)	0.851 (0.012)

B. Word embeddings according to the noise distributions

Figure 8 shows the word embeddings for the pairs except for the DVD and Electronics pair in Fig. 3. All pairs have domain-separable word embeddings for domain-independent noise distribution and domain-inseparable word embeddings for domain-dependent noise distribution.

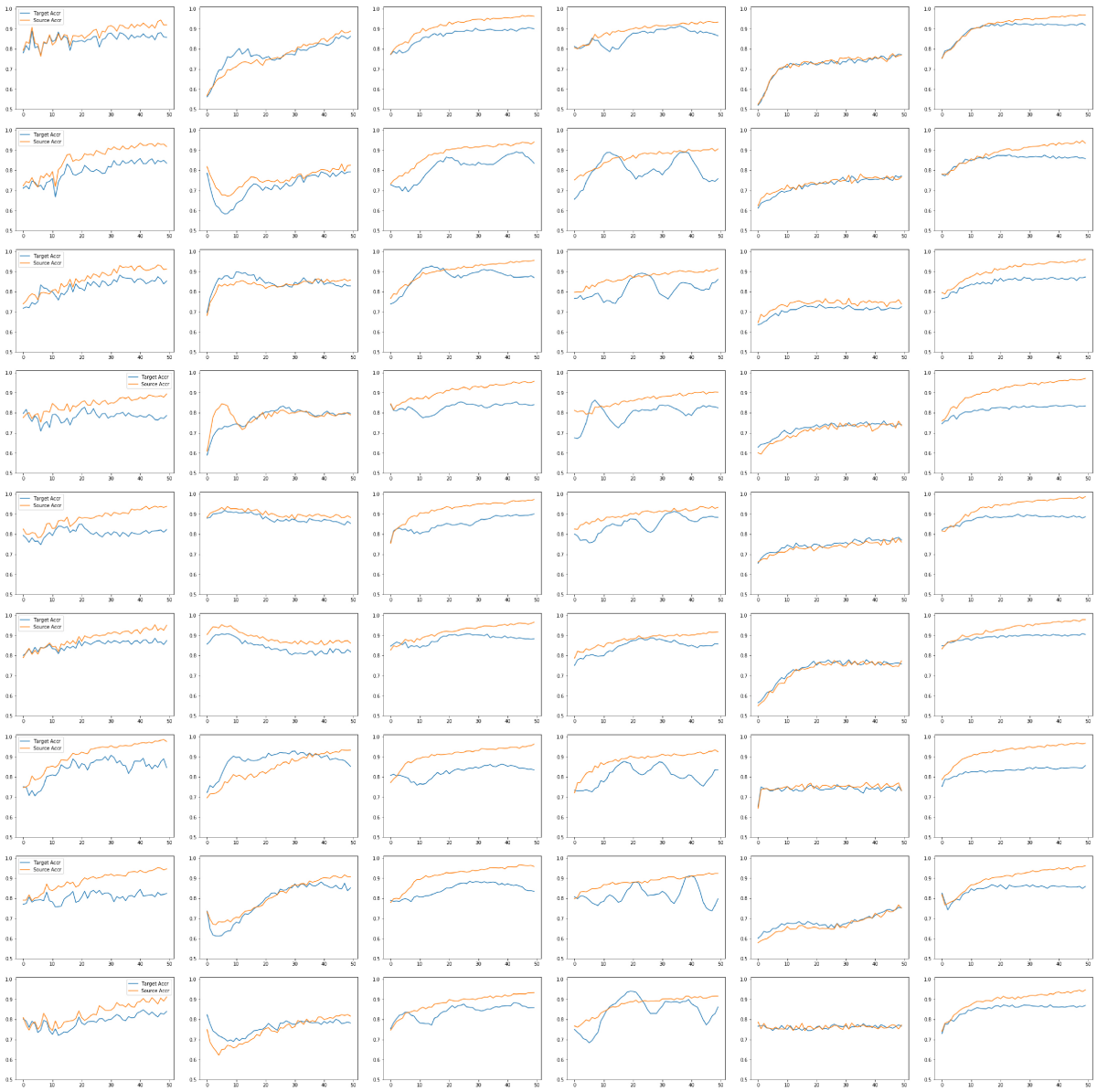


(a) Domain-independent noise distribution (b) Domain-dependent noise distribution

Fig. 8. Word embeddings from two different noise distributions where black dots represent common words, and black and blue dots represent domain-specific words.

C. DANN accuracy by epoch

Figure 9 shows the accuracies by epoch on the remaining pairs except for the pairs in Fig. 7 when training DANN. Figure 7 shows that our proposed methods have stably increasing patterns compared with other methods.



(a) BoW (b) TF-IDF (c) DBoW (d) DM (e) SCL (f) Proposed method

Fig. 9. Accuracy by epoch on document representations of Amazon dataset in experiments (Book → DVD), (Book → Electronics), (DVD → Electronics) (Electronics → Book), (Electronics → DVD), (Electronics → Kitchen), (Kitchen → Electronics) and (Book → Kitchen). Orange line and blue line refers to source training accuracy and target training accuracy respectively.