EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Cross-domain sentiment classification initiated with Polarity Detection Task

Nancy Kansal[1,*], Lipika Goel[1] and Sonam Gupta[1]

[1]Ajay Kumar Garg Engineering College, Ghaziabad, Uttar Pradesh, India

Abstract

INTRODUCTION: The requirement of the labeled dataset in the source domain makes the Cross Domain Sentiment Classification (CDSC) task complicate in the situation when the dataset is labeled manually.
OBJECTIVES: To overcome the dependency of CDSC tasks on manual labeling of the dataset by proposing a polarity detection task.
METHODS: We have proposed the CDSC-PDT method that is the polarity Detection Task (PDT) followed by the CDSC task. The proposed PDT task extracts the polarity of reviews from the source domain using the contextual and relevancy information of words in documents and this automatic labeled dataset is further used to train classifiers to make the further classification.
RESULTS: Proposed method is comparable to the traditional learning method giving the highest precision 85.7%.
CONCLUSION: The proposed method does not need to manually label the documents in either of the domain (source or target), hence it overcomes the human intervention and is also time saving and cheap process, unlike traditional CDSC tasks.

Keywords: Machine Learning, Sentiment Analysis, Polarity Detection Task (PDT), Cross-Domain Sentiment Analysis.

Received on 28 May 2020, accepted on 10 August 2020, published on 11 August 2020

doi: 10.4108/eai.26-5-2020.165965

1. Introduction

Sentiment Analysis is an evolving application of natural language processing (NLP). With the up-gradation of web 1.0 to web 2.0, the opportunities have also emerged for the service users to be active on these websites. With the advancement of the online mediums (like E-commerce, E-banking, Social Media) provided to users, Sentiment Analysis has also become an evolving research area as this is the only way to make people believe on the digital services (like e-shopping, music, movies) provided by these online mediums. Sentiment analysis analyzes the preferences and opinions of people (given in the form of online reviews or tweets) who have used the services, and tells their orientation, either positive or negative. This analysis for orientation (polarity) prediction can be used to make various decisions for, the users: to use the service or to buy any product, as well as to the service providers: to make decisions based on users' predilections that can be applied to make recommendation systems.

Various researches have been carried out for Sentiment analysis and have also successfully attained acceptable results

[*]Corresponding author. Email: Er.nancy001@gmail.com

performed on different datasets like IMDB movie reviews dataset (Duan et al., 2018) [4] and Amazon product reviews dataset (Blitzer et al. 2007) [1] that are used for training the models and these trained models are supplementarily used to classify the unlabeled opinions (reviews) in positive or negative labels. Meanwhile, models that are trained to accomplish this task need labeled dataset which is an extremely time consuming and dear process as labeling is done manually. Hence, in pursuance of attenuating this limitation of the labeled dataset, the alternative is to apply knowledge from one domain to make predictions for a different domain. And that is what is called **Cross Domain Sentiment Analysis (CDSA).**

Cross-domain Sentiment Classification (CDSC) is the task to classify (in positive or negative class) the written thoughts, attitudes, sentiments, or opinions of any person in one domain with the help of the model that is trained using the labeled written documents in different domain. Sentiment Analysis can be done by considering the users' opinions into three ways: by document, by sentence, and by aspect (Zhou et al. 2019) [25]. Document and sentence based sentiment analysis task conjecture the sentiment as either positive or negative. However, this may not be reasonable sometimes as a document or sentence may contain various aspects and hence various sentiments or attitudes of writers related to them. In the sentence and document-based sentiment analysis, sentiment orientation is predicted by considering sentence and document as a whole. Several pieces of research have been performed for the CDSA task in the last few decades. However, most studies have been effectuated on document-based Sentiment Analysis. A document (i.e. a review, tweet, etc.) may contain different sentiments for different aspects that may be concealed in the sentiment for the document as the whole. For example, a review posted on Amazon about the newly launched Apple's iPhone 8 64 GB, that is: "*This phone is great and the price is fantastic. Sadly there's an issue with the power button*" shows the positive sentiment about the phone but is showing negative behavior towards the power button of the phone. Hence, when contemplating the overall document, aspect level sentiments' orientation is lost. Various related researches have been performed in CDSA for Aspect level sentiments' analysis. Yang et al. (2019) [23] proposed an attention mechanism for identifying various aspects. They proposed Neural Attentive Network for aspect level CDSA tasks. Tang et al. (2016) [19] proposed deep memory networks to apprehend the significance of the words' context. Document-level sentiment analysis is an emerging area in the field of cross-domain sentiment analysis. Document-level CDSC tasks that have been performed over the last few decades needs the manually labeled dataset in at least one domain to perform CDSC task, as the labeled dataset is used to train the Machine learning classifiers and then this trained model is used to perform the classification task on target domain's unlabeled documents. Although various techniques are applied to the datasets to extract important features that contribute to classification task and different preprocessing steps are applied to datasets to enhance the CDSC task, yet the requirement of the labeled dataset (even in the single domain) sometimes become the threat to CDSC task, as the labeling is done manually, which is very time consuming and costly process. This study aims to focus on this labeling aspect of one domain to perform the CDSC task. We have built a **Polarity Detection Task (PDT)** that aims to detect the polarity of every document in the source domain. Instead of using the manually labeled dataset, we tried to extract the polarity of documents using words' importance and context information in the given domain's documents. This PDT task is followed by sentiment classification task of documents in different domains using the baseline machine learning models: Multinomial Naïve Bayes (MNB), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM) and Logistic Regression (LR). The baseline methods with PDT task are called as MNB_PDT, SGD_PDT, SVM_PDT, and LR_PDT. These models are trained using the source domain data along with the labels extracted from the PDT task and then the trained models are used to classify the sentiments/documents in the target domain.

Our study aims to give answers to the following research questions:

Ques 1: Is this type of PDT task feasible to perform a CDSC task?

Ques 2: Is the proposed model comparable to the CDSC tasks that use manually labeled datasets?

These questions are answered using the experimental results of our proposed technique/method to perform the CDSC task. We will be using polarity and orientation words interchangeably throughout the writing of this paper.

Rest of the paper contributes the following work to our research process:

Section 2 reports some piece of related works that are contributed to execute the CDSC task over the last decade. Section 3 illustrates the proposed modeling. Section 4 denotes the proposed methodology that describes the flowchart for all the steps performed to conduct the required CDSC task. Section 5 gives the experimental results to check the feasibility of the proposed method and to give the answers to the stated research questions (Ques 1 and Ques 2). Section 6 concludes the paper and gives some future directions to enhance the PDT task to improve the CDSC performance.

2. Related Work

A piece of research work that has been performed to accomplish the CDSC task, is analyzed to give a short description of studies.

Shared knowledge learning is proposed by Duan et al. (2018) [4] to address the problem of the large labeled corpus in all domains. They used bi-GRU combined with an Adversarial network to extract shared domain-independent knowledge in multi-domains in the shared knowledge learning part and in the shared knowledge transfer part, they transferred this knowledge to target domain for domain adaptation.

Pan et al. (2010) [16] proposed SFA (Spectral Feature alignment) that divided the features into two parts: domain-independent (pivots) and domain-specific features (non-pivots). It used a bipartite graph to fill the gap between these pivots and non-pivots. In this paper, they trained a binary logistic regression model. Li et al. (2012) [27] proposed a new bootstrapping-based method; they proposed Relational Adaptive Bootstrapping (RAP) for expanding the lexicon to retrain the classifier. Transfer Adaboost learning (TrAdaBoost) algorithm (Dai et al., 2007) [29] is used for learning in RAP. They have used SVM as a base classifier in Tr-AdaBoost.

Bollegala et al. (2013) [2] presented sentiment sensitive thesaurus that measured the similarity between two words and expanded features vectors with additional related elements to overcome the feature mismatch problem. Thesaurus is created to expand feature vectors during training and testing of L1 regularized logistic regression-based binary classifiers both for in-domain and cross-domain and always find state-of-the-art improvement. In-domain accuracy was more than cross-domain. Bollegala et al. (2016) [3] proposed a new model for CDSC tasks using sentiment sensitive embeddings.

Heredia et al (2016) [5] were very first to examine the effect of cross domain sentiment analysis on tweets and reviews. They conducted 18 experiments on two tweet datasets and one review dataset by varying training and testing datasets and classifiers. They performed these experiments on three classifiers: Naïve Bayes (NB), Multinomial Naïve Bayes (MNB), and Support Vector Machines (SVM) and found the state of the art results on MNB using tweets and reviews as training and testing datasets respectively.

All the models were focusing on domain-independent features, and then domain-specific features were also considered in later studies. Ganin et al. (2016) [30] proposed DANN (Domain Adversarial Training of Neural Network) for Cross-Domain Sentiment Analysis. DANN uses domain-independent information by ignoring domain-specific information**.** (Manual selection of pivots)

To enhance this, Pan et al. (2017) [31] proposed an end-to-end Adversarial Memory Network (AMN). It has a capability of automatic capturing of domain-independent words called pivots using the Attention Mechanism. It has two networks: one for Sentiment Classification and another for domain classification. Then joint learning was applied on both networks. AMN cannot capture non-pivots automatically. However, non-pivots (Domain-specific features) can play an important role in Sentiment Classification. (Automatic capturing of pivots)

Hence, Li et al. (2018) [7] proposed the Hierarchical Attention Transfer Network (HATN) for CDSC. It consists of two networks: P-net and NP-net. P-net performs attention learning to find positive and negative pivots. These are used as a bridge to find non-pivots. Then, P-net and NP-net conduct joint attention to find pivots and non-pivots and transfer attentions for emotions in different domains. HATN model has some negative points that were its increasing complexity due to the labeling of pivots and inevitable labeling error (like the word ‘sad’ was having a positive label in both kitchens as well as video domain).

To overcome these problems, T. Manshu and Z. Xuemin (2019) [10] proposed an end-to-end CCHAN model for the CDSC task. CCHAN consists of two networks: one called CTN and another called CHAN.CTN for prediction masked words. CHAN used cloze tasks for masking of words. They used 3-layer CNN in Hierarchical Attention Network for CDSC. They also used a matched degree between a document and answers. Tu Manshu and Wang Bing (2019) [9] enhanced the HATN model by introducing the SDM layer to extract more important words for sentiment analysis. They used sentiment dictionary as prior knowledge thus called it HANP (Hierarchical Attention Network with prior knowledge). They used 3-layer CNN to preserve contextual information from source to target domain.

Zhang et al. (2018) [24] very first used capsule networks for CDSC task to overcome the representational limitations of Recurrent Neural Networks (RNN) and Convolution Neural Networks (CNN). They proposed Capsule Networks in Domain Adaptation Scenario using Semantic Rules (CapsuleDAR) which was having two networks: a rule network and a base network. Rule network was used to leverage semantic rules into capsule network, a similar structure to this, the Base Network was the capsule network that was trained with features to perform sentiment prediction. Yang et al. (2019) [23] proposed NAACL for domain-specific aspect level sentiment analysis. It uses a weekly supervised Latent Dirichlet Allocation Model (Wilda) to learn Domain-specific Aspect and sentiment Lexicon representations. Aspect level sentiment classifier uses domain classification results and aspect document representation to classify aspect level sentiments in the target domain. LSTM is used to encode the input document. NAACL transforms document embeddings to domain-specific document embeddings

Meng et al. (2019) [12] have used word2vec for finding word embeddings at the embedding layer and then those embeddings are inputted to a 3-layer CNN and sentiment prediction is done using softmax classifier. They have shared the weights of the convolution layer and max-pooling layer in source and target domain samples and fine-tuning is done on the fully connected layer of proposed CNN. Natalia Ponomareva and Mike Thelwall (2013) [17] proposed the modified version of the Label Propagation (LP) algorithm developed by Zhu and Ghahramani (2002) [22] to compare the performance of semi-supervised learning (SSL) and cross domain leaning (CDL) using graph-based algorithms. They also check the consequences of graph structure modified using parameter variations**.**

Akthar et al. (2016) [28] proposed a MOO (Multi-Objective Optimization) framework that was used to select some optimized features that were augmented with the feature vectors that were learned using CNN. They were very first to apply the proposed model for the Hindi language dataset. Wei et al. (2017) [20] proposed the cross-domain semantic correlation auto-correspondence method (CSCW). They used semantically invariant word features and then they pull out common frequently occurring top pivot features from source and target domains. They used word2vec to find the semantic similarity between the pivot and non-pivot features and extracted the feature pairs that reveal similar opinions but were having different representations and align these pairs into similar feature representations to train the SVM classifier. Meng et al. (2019) [12] proposed CNN for transfer learning in the CDSA task. Firstly, they used word2vec to find word embedding at the embedding layer and then train a 3-layer CNN. They applied the Softmax classifier for polarity generation. They shared the weights in two layers of proposed CNN and fine-tuned the weights at the last layer of CNN.

Although studies have used the outstanding frameworks to perform efficient CDSC tasks, yet near all are dependent on manually labeled datasets to perform CDSC tasks. To the best of our knowledge, no study has yet performed PDT task prior to the CDSC task in their study. Hence, the proposed method is just the initiative to the advancement towards the CDSC task by overcoming the dependency on manual labeling of datasets.

3. CDSC-PDT Modelling

This section introduces the problem definition followed by the details of components of the proposed CDSC-PDT model.

3.1. Problem Interpretation

Let the two domains for any CDSC task are represented as $D(s)$ and $D(t)$ for source and target domain respectively. Each domain has a set of documents (1600 in the source domain and 400 in the target domain). Every document in the source domain d^s is the collection of n sentences $\{s_i\}_{i=1}^{n}$ and every sentence is the collection of m words$\{w_{ij}\}_{j=1}^{m}$. The main aim of the proposed method is first to find the labels of unlabeled documents of the source domain and second to train the Machine Learning classifier using these drawn labels for source domain documents and then test the trained model to find the orientation of documents for target domain dataset.

3.2. Proposed framework

The proposed CDSC-PDT model framework is shown in figure 1. Each document in the source domain contains n sentences$\{s_i\}_{i=1}^{n}$. And each sentence is the collection of m words$\{w_{ij}\}_{j=1}^{m}$.

Word Level Representation Layer (WLRL)

Each word from every sentence is first represented into its embedding representation $\boldsymbol{e_{ij}}$using word2vec. These word embeddings are vector representations of words such that similar context words are assigned similar representation ($\cos\theta = 1$), near similar words are assigned similar representation ($-1 < \cos\theta < 1$) and opposite context words are assigned different representation in vector space ($\cos\theta = -1$). Additionally, each word also has to maintain its importance in every document. Hence, each word is represented in vector form using TF-IDF. The vector representation thus extracted $\boldsymbol{\delta_{ij}}$ is the tf-idf score of each word in every sentence.

Clustering Layer (CL)

The embeddings of word obtained from word2vec $\boldsymbol{e_{ij}}$ **(**that is the vector representation of words assigned to them based on their context**)** are fed into the unsupervised k-means clustering model that groups the words into two clusters based on their Euclidean distance (closeness score) of each word from the cluster centroid. Thus in obtained clustersC_0 *and* C_1, the positive words are in C_0 and negative words are inC_1. The closeness score of each word is multiplied by +1 and -1 for C_0 *and* C_1 respectively. Thus the weighted orientation score for each word is calculated by taking the inverse of their +ve and –ve closeness score.

Thus, the output of the clustering layer is the word orientation dictionary O_w for each word of every document which has the weighted orientation score $\boldsymbol{o_{ij}}$ for each word of every document.

Document Level Representation Layer (DLRL)

At this layer, all words in every document are one hand supplanted with their weighted orientation score $\boldsymbol{o_{ij}}$**(**output from clustering layer**)** and supplementarily with their tf-idf scores $\boldsymbol{\delta_{ij}}$ (output from word-level representation layer).

Thus the output of DLRL is the two vector representations of every document.

Document Orientation Dictionary Layer (DODL)

In order to make our CDSC task more effective we have combined the importance of word with the semantic meaning of words that we have combined the tf-idf scores (representing word importance) with the word2vec scores (representing semantic meaning) of words. For this, we calculated the dot product of the two vector representations of every document output from DLRL. The dot product thus achieved was used to perform **The PDT** for every source domain data for each CDSC experiment. As the document with –ve dot product was assigned the label 0 and the product with +ve dot product was assigned the label 1.

Thus, the output of this layer is the Document Orientation DictionaryO_{d^s}. Which contains the sentiment polarity of each document in the source domain.

Training Level (TL)

Once the document orientation Dictionary O_{d^s} is found, the Machine Learning classifiers: MNB, SVM, SGD, and LR are trained using the extracted labels of the source domain dataset. And this trained classifier is used for the next-level testing process.

Testing Layer (TeL)

At this level, the trained classifier is tested for unlabeled documents d^t for target domain D (t) to find the orientation of sentiments from these documents. Target domain document is represented as the collection of N sentences$\sum_{i=1}^{N} s_i^t$. And every sentence is a collection of M words$\sum_{i=1}^{M} w_i^t$. The features are extracted from target domain documents based on the word importance in documents by using the TF-IDF feature extraction method. Thus, the vector representation of documents δ_{ij}^t are fed into trained classifiers to predict the polarity of target domain documents. Hence, the output for this layer is the label for every document of the target domain.

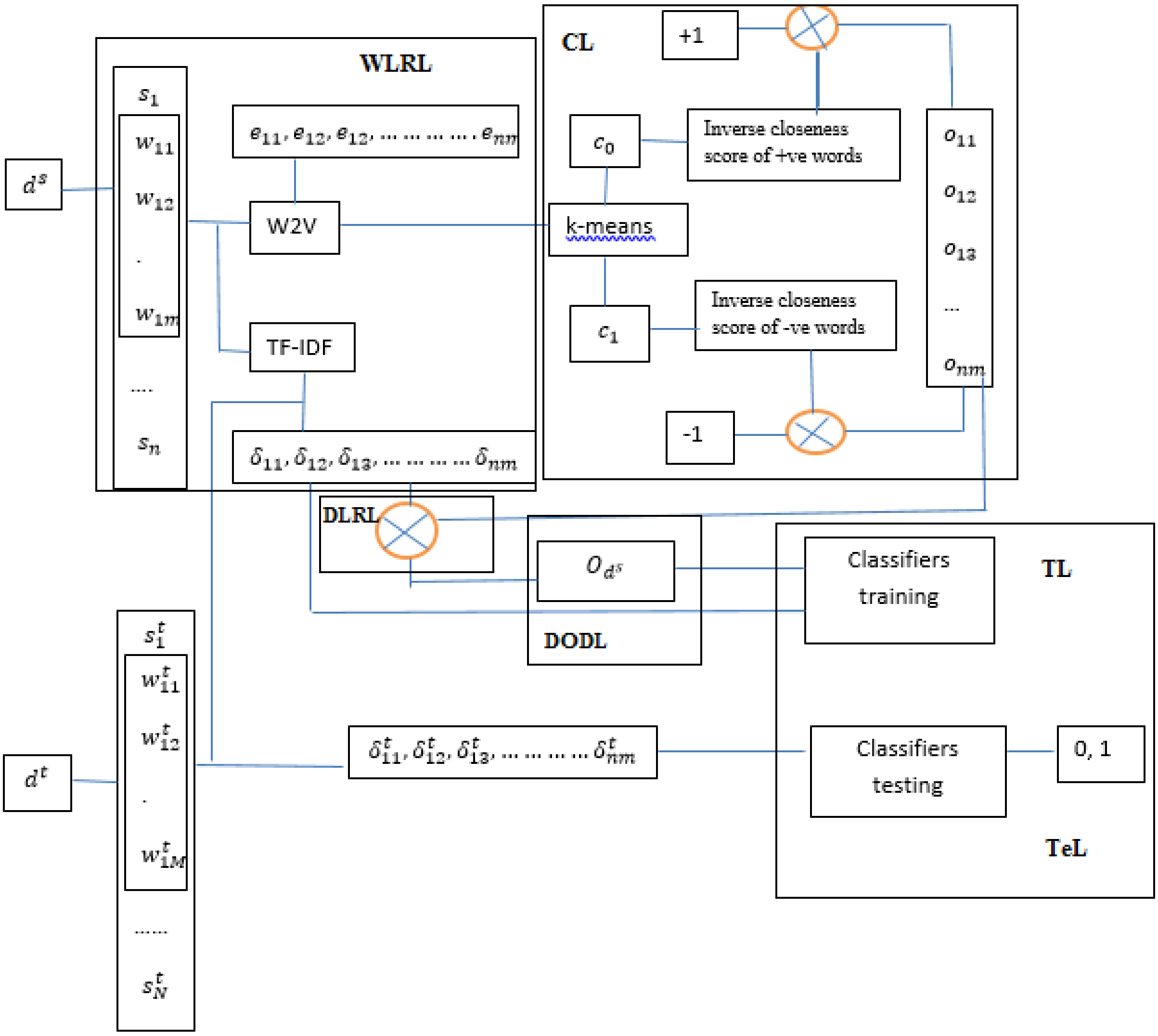
WLRL
s_1
w_{11}
w_{12}
.
w_{1m}
....
s_n
d^s
$e_{11}, e_{12}, e_{12}, \dots \dots \dots \dots, e_{nm}$
W2V
TF-IDF
$\delta_{11}, \delta_{12}, \delta_{13}, \dots \dots \dots \delta_{nm}$
CL
+1
c_0
Inverse closeness score of +ve words
k-means
c_1
Inverse closeness score of -ve words
-1
o_{11}
o_{12}
o_{13}
...
o_{nm}
DLRL
O_{d^s}
DODL
Classifiers training
TL
s_1^t
w_{11}^t
w_{12}^t
.
w_{1M}^t
......
s_N^t
d^t
$\delta_{11}^t, \delta_{12}^t, \delta_{13}^t, \dots \dots \dots \delta_{nm}^t$
Classifiers testing
0, 1
TeL

Figure 1. Model Framework

4. Proposed Methodology

Figure 2 shows the proposed model that has been used to perform the desired CDSA task. The step by step process is explained following.

4.1. Data Collection/Dataset

Amazon products reviews dataset (Blitzer et al. 2007) [1] is the most widely used dataset for CDSC tasks. It consists of product reviews from four domains: Books, DVDs, Kitchen, and Electronics Appliances. The reviews are labeled based on ratings in such a way that the reviews having rating <3 are labeled as negative and that has >3 are labeled as positive. The 4 products may be considered as 4 domains and various studies [2, 3, 4, 5, 7, 9, 10, 16, 24] have been proposed using this dataset.

We have taken the Amazon reviews dataset (Blitzer et al., 2007) [1] as this is widely used to perform the CSDC task. The motive behind taking this labeled dataset is that we have to compare the results of our proposed method, that does not take manually labeled dataset but it withdraws the labels from the information that is provided in reviews, to some baseline methods used to perform CDSC task on labeled source domain dataset. The dataset has 2000 reviews for each of the 4 domains: Books, DVD, Electronics, and Kitchen. Table 1 shows the interpretation of reviews of the Amazon product reviews dataset.

4.2. Preprocessing

Figure 3 shows the preprocessing steps used to accomplish the desired CDSA task. Following preprocessing steps are executed on the source as well as the target dataset to carry out the proposed CDSA task.

- Relinquish rows having lost (NaN) values,
- Relinquish two or more identical rows,
- Remold all uppercase into lowercase letters,
- Relinquish stop words,
- Supplant all non-alphabetic words with a single white space,
- Stem the words to their root words.
- Bigrams are extracted from the corpus of words taken out from the source domain's reviews.

No of Bigrams extracted from all the domains are discussed in table 2.

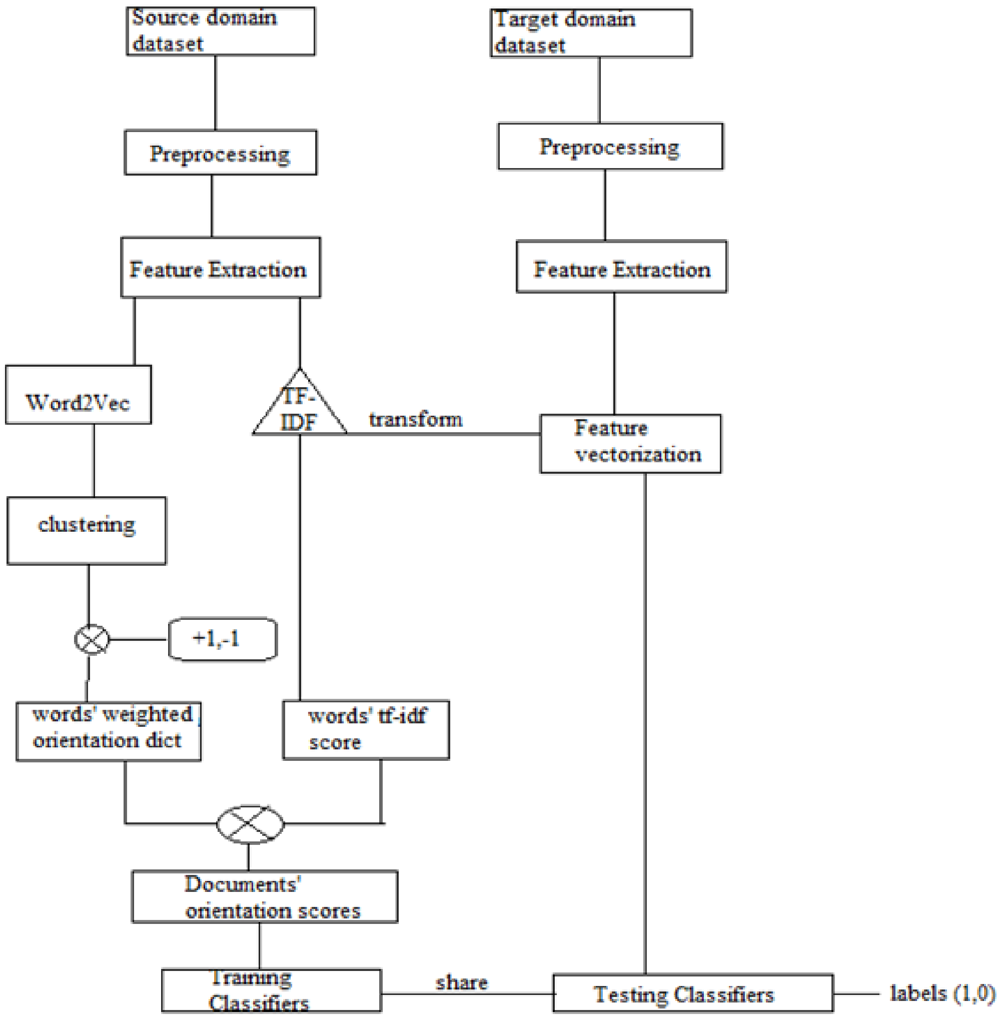

Figure 2. Proposed model

Table 1. Dataset Interpretation

Dataset	Domain	Positive reviews	Negative reviews	Total
Amazon product reviews	Books(B), DVD (D), Electronics (E), Kitchen (K)	1000	1000	2000
		1000	1000	2000
		1000	1000	2000
		1000	1000	2000

Table 2. Bigram extraction

Domain name	Source vocab length (word types)	Words Corpus length (Unigram+Bigram)	Extracted Bigrams

Book	128904	213980	41519
DVD	125617	314636	42598
Electronics	74476	120048	16957
Kitchen	61911	93761	12472

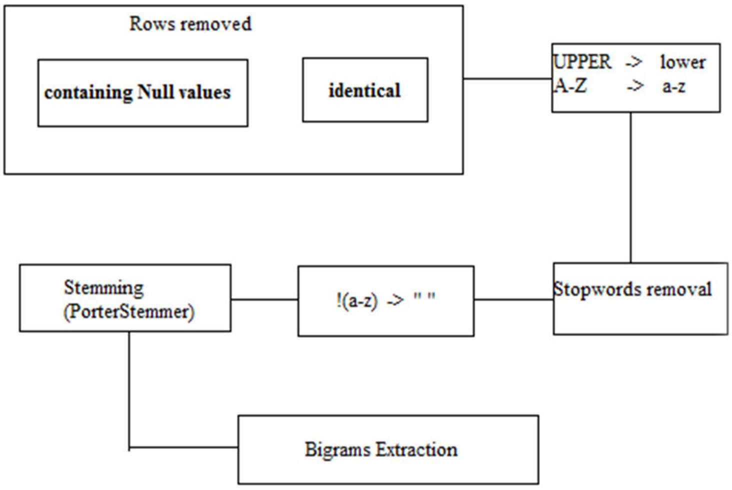

Figure 3. Preprocessing steps

4.3. Feature Extraction

Various Machine learning algorithms are used for CDSC tasks. The only condition to use these algorithms is that they do not work on categorical data (also called features). Hence all the data, before analysis (using these algorithms), are first converted into numerical form. Thus the process of converting the features into vectors is called feature Extraction. Sentiments are sentences that are a group of words and these words are first encoded into numerical form, also known as vectors before they are feed into a machine learning model for training as well as the testing process. Various methods can be used to represent the text into numerical form. We have discussed some of the methods. The following methods are used in our study for vectorization of text to perform the CDSC task.

TF-IDF

TF-IDF is a feature extraction technique that converts the features into vectors by their weights of importance [26]. (Chris Albon, 2018). TD-IDF encodes the text based on the significance of words in documents (reviews). It gives the output features (vector representation) weighted by their significance to the document. It compares the frequency of the word in the document (reviews or tweets) and compares it with the frequency of that word in other documents as well. It calculates the tf-IDF scores of words in such a way that:

If a word occurs most frequently occurred in a document but does not occur in other documents means that word is relevant to that document only. This is called word frequency or term frequency (TF) [26] (Chris Albon 2018). And it can be calculated as:

$$\boldsymbol{TF_{t,d} = \frac{t_d}{n_{t,d}}} \tag{1}$$

Where, $TF_{t,d}$= term frequency of the term t in document d

t_d= number of times t occur in document d

$n_{t,d}$= number of terms in document d

A complement to this, if a word occurs frequently in many documents means that word is not relevant for any particular document. This is called the Inverse Document Frequency (IDF) (Chris Albos 2018). And it can be calculated as:

$$\boldsymbol{IDF_t = log\frac{N}{N_t}} \tag{2}$$

Where N= total number of documents

N_t= Number of documents having the word t

The value of IDF varies from 0 to 1 most commonly. This denotes that the word is not relevant if this score is 0 and is most relevant if the score is 1. Relevancy is moderate in between 0 and 1.

These two statistics are multiplied together to find the tf-idf score to assign to every word in the document.

$$\boldsymbol{TF\_IDF_{t,d} = TF_{t,d} * IDF_t} \qquad (3)$$

We have used the *TF-IDF vectorizer* to encode source as well as target domain reviews into numerical form. This returns the m*n matrix representation of each document. Where m is the number of documents and n is the number of words in all the documents that contribute to polarity detection based on their uniqueness in the documents. And the matrix is having a tf-idf score corresponding to each word in every document. We have considered the feature vectors individually to contribute to the overall document orientation finding.

Word2Vec

Word2vec (Mikolov et al., 2013a, 2013b) [13, 14] is another method for finding the word embeddings. Word embeddings represent the words into vector form by preserving their contextual information and their semantic and syntactic similarity with other words. It transforms the words in such a way that if the contextual meaning of two words is the same, their vector representation should also be spatially close to each other. This means that the cosine angle between two such vectors will be close to 0. And if the words are opposite in context their vectors should be spatially opposite to each other. And the cosine angle between the two vectors then is 180.

Cosine similarity between the two vectors X and Y, which are the word embeddings of the two words w1 and w1 will be denoted by:

$$\boldsymbol{Cosine\_simmilarity_{w1,w2} = cos\Theta = \frac{X.Y}{|X|.|Y|}} \qquad (4)$$

If $\boldsymbol{\Theta}$=0, w1, and w2 are most similar,

If $\boldsymbol{\Theta}$=180, w1, and w2 are opposite in meaning.

An important characteristic of word2vec is that it is independent of the orientation of sentences that are feed into it to find the word embeddings for every word of the sentence.

Word2vec uses two methods to find the word embeddings of words: CBOW (continuous bag of words) and the Skip-gram method. Given the contextual information of words, the CBOW method predicts the word. And when given the word, the Skip-gram model predicts the contexts of the words. We have used the Skip-gram method of word2vec that takes the neighboring words as an asset of input and predicts the middle words based on contextual information of the neighboring words by considering them as labels. It does not need the manual labels of the documents. Hence, this is considered as unsupervised to implement our proposed method.

As word2vec gives the vector representation of words that can be used for machine learning, it is also called a feature extraction method. This means that if we would have movie reviews dataset, word ‘boring’ would be surrounded by the same words as word ‘tedious’, and usually, such words would have somewhere close to the words such as ‘didn’t’ (like), which would also make word didn’t be similar to them., and according to Word2Vec they will, therefore, share a similar vector representation. These word representations of words can be used to perform various tasks like clustering and classification of textual data.

We have used genism’s implementation of word2vec to find the word embeddings for the extracted Bigrams. Word2vec is used to assign word embeddings to the features/words in such a way that the words that are having similar context in documents of respective domains are given similar embedding/vector representation. The contextual Similarity of these words is calculated based on the cosine similarity discussed in equation 4.

4.4. Clustering

K-means clustering (MacQueen J., 1967) [8] is an unsupervised machine learning algorithm that is used to cluster input data based on their similarity and distinctness. The features that are similar to each other are grouped in a single cluster are that are distinct, are grouped in different clusters based on the similarity criteria. K-means clustering has wide usage in the research area by virtue of its simple and swift behavior. It follows a repetition mechanism to group data into k clusters. Each of the clusters has a cluster center also known as cluster centroid. The centers of clusters are initialized in starting. These are called seed points of the clusters. Euclidean distance is calculated thereafter for each input with feature with respect to the centroid of each cluster and based on the calculated distance, the feature is assigned to the cluster having the least Euclidean distance with its center. If there are n input features: $a1, a2, a3, \ldots \ldots . an$ and we have to group them into k clusters $c1, c2, c3, \ldots \ldots ck.$ then, the distance can be calculated as:

$$\boldsymbol{X_{a,c} = \sum_{i=0}^{n} \|a_i - c_i\|^2} \qquad (5)$$

Where, $X_{a,c}$= Euclidean distance between the feature vector a and cluster center c

a_i=vector representation of ith input feature

c_i= centroid value of the ith cluster

Features are assigned to the clusters based on the following condition (S). Following condition must be satisfied by a_r feature to be assigned in the i^th cluster.

$$S(a_r, i) = \begin{cases} i, & i = \arg\min\left(\| a_r - c_j \|^2 \quad j = 1,2, \dots \dots \dots . k\right) \\ 0, & otherwise \end{cases} \tag{6}$$

Features are assigned to different clusters based on this Euclidean distance, the less the distance the more the probability of the feature to be assigned in the current cluster.

Following steps are taken to group the features into k clusters:

Step 1. Initialize the number of clusters k, to perform the desired clustering.

Step 2. Initialize seed points for all k clusters. Diverse methods can be used to initialize cluster centers. The generally used method is the Random selection method, in which k random numbers are assigned as seed points for k clusters.

Step 3. Membership of features in k-clusters cab be calculated based on the Euclidean distance $X_{a,c}$ formula denoted by equation 5 to decide which feature is closest to which cluster centroid.

Step 4. Once the features are assigned to any cluster, the center of the cluster is recalculated as the mean or median of all feature values belonging to the current cluster.

Step 5. Repeat steps 3 and 4 for all features until all features are assigned to any of the clusters and all the clusters became inconsistent.

In our experiment, k-means clustering is used to cluster the features in such a way that the features that are having less distance between their word vectors/embeddings are grouped into one cluster. Euclidean distance denoted by equation 5 is used to calculate the distance between the two feature vectors and the features are assigned to clusters based on the condition discussed in equation 6. We have taken the number of clusters to be 2 for our classification task, one denoting positive cluster and other denoting negative cluster based on the feature assignment to both clusters. This returns the Euclidean distances of features with respect to their cluster centroids. And either of the 2 cluster labels of any feature denotes the orientation of the word (either positive or negative) to contribute to overall document polarity.

4.5. Words' weighted orientation

As the words are taken as the basic unit to construct the word embeddings and these words are clustered, hence the cluster labels of each word denote the orientation of words to be positive or negative. The clustering results return the weighted orientation score of words that is the product of the inverse closeness score (distance) and cluster labels. This score is used to find the document vector representation by replacing each word in the document with its corresponding weighted word orientation.

4.6. Document orientation

The uniqueness of every word can be combined with the context information of that word. Hence, the two vector representations of every document that are obtained using words' weighted orientation and tf-idf, are combined to get the overall orientation of the document (review) and hence resulted in source domain data labeling **(Polarity Detection Task)**.

4.7. Learning and Testing

The obtained documents' orientation can now be utilized to train the following 4 machine learning classifiers/algorithms for sentiment classification/Analysis task.

Multinomial Naïve Bayes (MNB)

It calculates the likelihood of the occurrence of any feature vector in a particular class. It generates the multinomial distribution of the probability of any document occurring in class 'c' (0 or 1 for our classification task) (McCallum et al., 1998) [11].

Support Vector Machine (SVM)

Support vector machines introduced by (Vapnik and Vladmimir, 1995) [21], draw a hyperplane between the two classes of feature vectors when represented in 2-D space. It draws the hyperplane in such a way that there may be the maximum distance between the feature vectors of both the predicted classes (that are positive and negative in this CDSA task).

Joachims, T. (1998) [6] has developed multiple variants of SVM. Linear SVM is used mainly due to its popularity and high performance in text classification.

Logistic Regression (LR)

Logistic Regression is a statistical model that calculates the variation of the predicted class label from the actual class label by calculating the cost function. This cost function is the distance between the predicted class label and actual class label (Omurlu et al., 2008) [15]. A Binary logistic regression model has two output values (positive and negative in our study). It calculates the probability for any feature to occur in any class in terms of its input value, that probability may be compared with any threshold value as the boundary between the class labels such that below that threshold the label may be assigned to 'negative' or '0' and above which the label may be assigned as 'positive' or '1' and vice-versa.

Stochastic Gradient Descent (SGD)

Stochastic gradient Descent is a resolver that uses gradient descent for minimizing the cost function. Gradient Descent is the iterative process that makes iterations for reaching its minimum cost function. In SGD a few random features are selected for each iteration process.

For each of the given training pairs$(d_1, y_1), (d_2, y_2), \dots\dots\dots(d_n, y_n)$, where, d_i and y_i are document and respective polarity respectively, It learns the linear function $f(z) = wz + b$ with normal model parameters, where the sign of $f(z)$ tells the predicted polarity of documents. We have used the SGD optimization algorithm with "*log"* loss in our experiments that results in the LR fitted with the SGD algorithm to solve the classification problem.

Thus obtained learned model can be used to perform testing for target domain sentiments' classification. We have performed the comparison of our proposed method with the models that are trained using manually labeled dataset so that we can compare the feasibility of this method (in which no manually labeled sentiments are used to train the classifier but the labels are extracted from the contextual and relevancy information of words provided in the document itself) with the previously proposed supervised CDSA methods (in which the CDSA task is performed with the prior requirement of the labeled dataset, thus the classifiers are trained using these labels and then further testing is performed for classification).

4.8. Performance Measures

Performance measures are customarily the statistical assessment results to compute the efficacy of Machine learning models. Sammut and Webb (2011) [18] have given some terms to do the statistical assessment or to calculate the efficacy of models. This statistical assessment is conducted using the following computations in our experiment.

Accuracy

It can be calculated by dividing the true predicting results by the total results. In terms of the confusion matrix Accuracy can be calculated using the following formula:

$$\boldsymbol{ACC} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+FP+TN+FN}} \qquad (7)$$

Precision

Precision can be computed by fractionalizing the number of results that are truly predicted to be positive by the total number of predicted positive results. It can be calculated using the formula:

$$\boldsymbol{P} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \qquad \mathbf{(8)}$$

Recall

Recall can be calculated by dividing the number of results that are truly predicted to be positive by the total number of actual positive. It can be calculated by the following formula:

$$\boldsymbol{R} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \qquad \mathbf{(9)}$$

5. Experimental Results and Discussion

5.1. Parameter settings

Word2vec

The fundamental unit for our experiment is taken as a word. To start with, word vectors for each word was constructed. We have found 300-dimensional word embeddings for each word in each domain by training the word2vec model. To implement the skip-gram model of word2vec, the window size is set to 4 to look up the word embeddings for similar context words. The threshold value, for down-sampling of frequently occurring words, was set to 1e-5. We have selected 20 noise words to be drawn for each experiment. The learning rate for word2vec was initially set to 0.3 and as the training make headway, it was deteriorated to 0.0007.

Clustering

To classify words' orientation in positive or negative, the number of clusters (k) was set to 2 (one for positive and other for negative words' clustering). The algorithm was set to be executed with 50 repeated seed points, to hinder the learning task from presuming wrong initial seed points' coordinates. New features were assigned to the cluster, after 1000 iterations for each assignment task.

The rest of the parameters were set to default.

5.2. Experimental study

Polarity Detection Task Analysis

The proposed PDT task is performed on the source domain for every experiment. We have extracted the relevancy information of words in terms of tf-idf score of features extracted from documents and contextual information using word2vec.

Table 3 shows the number of features extracted using tf-idf for every source domain and the number of word embeddings extracted from documents using word2vec the size of embeddings was set to 300 dimensions.

Table 3. Features comprehension

Domain	No. of features extracted using TF-IDF	No of word embeddings using Word2Vec	The training time of word2vec	Effective words identified by Word2Vec/ no. of raw words

Books	33686	6529*300	0.76 min	268501 / **1836120**
DVD	13589	6366*300	1.03 min	518778 / **3077300**
Electronics	16021	3954*300	0.87 min	268501 / **1836120**
Kitchen	12810	3446*300	39.4 sec	201937 / **1451020**

Classification task results analysis

We have performed a statistical assessment of our model using Accuracy, Precision, and Recall scores described in section “performance measures”.

We have carried out 12*4 experiments for the CDSC task using 4 different domains’ reviews from the Amazon product reviews dataset. 12*4 experiments are created by varying source-target domain pairs and different baselines. We are motivated to take this dataset as we will be comparing our proposed method of CDSA to some of the existing methods: MNB, SVM, SGD, and LR that are, unlike our proposed method, trained and tested on this manually labeled dataset. This dataset was prepared by Blitzer et al. (2007) [1] by manually labeling the customer reviews based on product ratings, to conduct the Cross Domain Sentiment Analysis task.

Every domain dataset is divided into train and test set in such a way that for every domain 80% of the data are set as train data, and rest are set as test data.

In every train data, 50% of data are positively labeled and 50% of data are negatively labeled. We have used this labeling information only to do a statistical assessment of our experiments. The detailed description of the train and test data for every domain is explored in Table 4.

Table 4. Train-Test data description

Domain name	Train set	Test set	Total
Books	800 (p) + 800 (n)	200 (p) + 200 (n)	1000 (p) + 1000 (n)
DVD	800 (p) + 800 (n)	200 (p) + 200 (n)	1000 (p) + 1000 (n)
Electronics	800 (p) + 800 (n)	200 (p) + 200 (n)	1000 (p) + 1000 (n)
Kitchen	800 (p) + 800 (n)	200 (p) + 200 (n)	1000 (p) + 1000 (n)

In-domain classification results

We have used MNB, SVM, LR, and SGD as the baseline methods to conduct our experiments. These methods compute the results by performing supervised learning to perform the classification task. They can adapt the learning task from the source domain to the target domain directly. The standard approach applied to these classifiers is when training and testing task is performed using the same domain. This classification task is called in-domain sentiment classification task in our experiment as the classifier is first trained using one domain dataset and is tested for sentiment classification tasks for the same domain (using different train and test set). Table 5 gives the accuracy results of these baselines when in-domain sentiment classification is performed using a manually labeled dataset of the source domain. Table 5 gives the accuracy results of these baselines when in-domain sentiment classification is performed using our proposed method. The task is performed on discussed Amazon product reviews dataset having 4 different domains product reviews: Books, DVD, Electronics, and Kitchen.

Table 5. In-domain classification results using manual labeling

	MNB	SGD	SVM	LR
Books	0.780	0.768	0.755	0.771
DVD	**0.820**	0.830	0.830	0.838
Electronics	0.818	**0.850**	**0.858**	**0.855**
Kitchen	0.813	0.815	0.810	0.815

Table 6. In-domain classification results using the proposed method

	MNB_PDT	SGD_PDT	SVM_PDT	LR_PDT

Books	0.663	0.682	0.665	0.676
DVD	0.701	0.731	0.750	0.741
Electronics	**0.800**	**0.813**	**0.787**	**0.780**
Kitchen	0.717	0.717	0.732	0.752

It can be seen from table 6 that results are best for the Electronics domain when Electronics is used as training as well as testing of classifiers in order to perform an In-domain sentiment classification task. Although the proposed 'PDT+ classification' task doesn't outperform the traditional learning process (when manually labeled data is used to train the classifiers), yet the proposed method is comparable to that.

Cross-domain classification results

To perform the CDSC task, we have made the domain pair in the form **U ||V** where U denotes the source domain and V denotes the target domain, And U and V belong to {Book, DVD, Electronics, and Kitchen}.

Table 7 shows the classification accuracy baseline classifiers using the traditional learning process to perform the CDSC task. The results show that the best classification is achieved for all baseline classifiers when the classifiers are trained using manually labeled datasets of the Kitchen domain and are tested for the labeling of documents in the Electronics domain. It is clear from Table 8 that when the PDT task is applied to extract the labels for the CDSC task, the standard results of baselines shown in table 5 have changed the scenario. It can be seen that all baselines show better accuracies for different CDSC tasks. Like, the MNB_PDT method gives better accuracy when DVD is used to perform PDT task and the results of these tasks (labeling) are used to train the MNB classifier, and then the trained classifier is used to predict sentiments orientations for Kitchen domain documents. Similarly, the SGD_PDT method gives better accuracy for Kitchen as source and DVD as target domain, SVM_PDT gives better results for Electronics as source and Kitchen as target domain, and LR_PDT gives better results for Book as source domain and Kitchen as the target domain. LR_PDT gives the highest precision that is 85.7% to perform the CDSC task when Electronics is used for training and the Book domain is used as a testing domain.

Table 7. CDSC results using traditional manually labels learning method

	MNB	**SGD**	**SVM**	**LR**
B || D	0.705	0.748	0.738	0.738
B || E	0.605	0.693	0.670	0.690
B || K	0.578	0.713	0.695	0.720
D || B	0.685	0.718	0.720	0.708
D || E	0.685	0.718	0.715	0.728
D || K	0.635	0.743	0.733	0.733
E || B	0.645	0.670	0.663	0.658
E || D	0.635	0.725	0.727	0.690
E || K	0.723	0.783	0.770	0.790
K || B	0.608	0.670	0.651	0.655
K || D	0.685	0.698	0.715	0.713
K || E	**0.733**	**0.818**	**0.805**	**0.805**

Table 8. CDSC results using the proposed strategy (label extraction using PDT task)

	MNB_PDT			**SGD_PDT**			**SVM_PDT**			**LR_PDT**		
	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.	Acc.	Pre.	Rec.
B || D	0.617	0.621	0.620	0.695	0.692	0.625	0.623	0.623	0.660	0.703	0.703	0.665
B || E	0.695	0.695	0.680	0.690	0.684	0.510	0.610	0.705	0.660	0.710	0.688	0.685

B || K	0.630	0.691	0.680	0.630	0.710	0.750	0.788	**0.750**	0.670	**0.770**	0.771	0.685
D || B	0.654	0.617	0.610	0.519	0.693	0.500	0.700	0.731	0.698	0.562	0.598	0.600
D || E	0.664	0.677	0.700	0.615	0.666	0.606	0.693	0.680	0.680	0.613	0.657	0.598
D || K	**0.777**	0.689	**0.750**	0.598	0.600	0.712	0.603	0.685	0.651	0.700	0.681	0.583
E || B	0.708	0.714	0.625	0.715	**0.766**	0.660	0.613	0.693	0.682	0.658	**0.857**	0.631
E || D	0.663	0.600	0.515	0.617	0.640	0.615	0.686	0.666	0.625	0.710	0.725	0.700
E || K	0.795	**0.777**	0.705	0.700	0.700	0.695	**0.792**	0.733	**0.715**	0.695	0.633	0.710
K || B	0.616	0.678	0.613	0.616	0.657	0.670	0.610	0.677	0.619	0.713	0.645	0.695
K || D	0.695	0.768	0.656	**0.785**	0.753	0.671	0.678	0.684	0.675	0.682	0.670	0.650
K || E	0.687	0.742	0.695	0.753	0.731	**0.741**	0.710	**0.750**	0.710	0.726	0.724	**0.755**

In order to give answers to the research questions discussed in section A, we have compared the results of baseline methods that are trained using the traditional learning process with the proposed PDT learning process to perform CDSC tasks. Figure 4 shows a comparison between the two methods. It can be seen from Figure 4 that although the proposed method doesn't outperform the traditional methods' accuracy, yet it is comparable to the traditional learning method and hence, able to answer our research questions (Ques1 and Ques 2).

From our experiments, it can be clearly stated that this initiation towards labeling of data using propose PDT method can give fruitful results in the cases when data can't be labeled manually due to time or cost.

The proposed method outperforms the human interaction aspect of traditional models which need manually labeled dataset in at least one domain (to be source domain) to perform CDSC tasks. And this study has tried to overcome this limitation by proposing the PDT task prior to the CDSC task. Hence, this approach can be found as a logical contribution to make CDSC task cheap, less time consuming, and less human intervention.

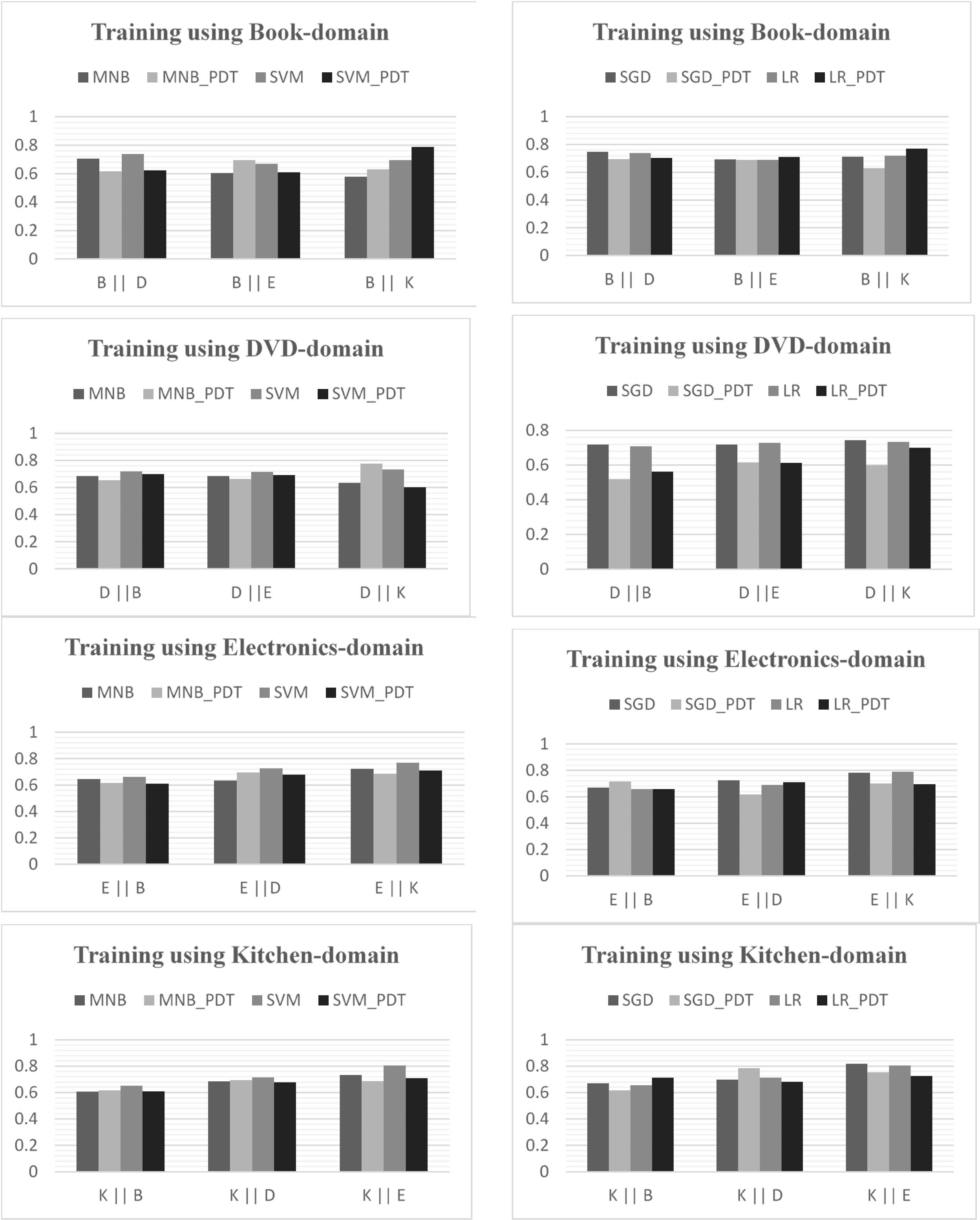

Figure 4. Comparison of PDT+ classification to the traditional learning process of baselines

6. Conclusion and future work

We are initiating the step towards the CDSA task where the manual labeling of documents is not needed but only the text data is required that may be in the form of reviews or tweets. From the experimental results, it can be concluded that the proposed method is comparable to traditional learning to perform the CDSC task. However, the proposed method does not need the manually labeled dataset in either of the source or target domain. The proposed method is a contribution towards the cheap CDSC task by lessening the human intervention and also time with is of great extent when the dataset is labeled manually by expertise.

The proposed study may be supplemented using various feature extraction tasks to perform PDT and CDSC as well. Word embeddings can be shared between domains and more precise word representations can be extracted. Other unsupervised machine learning models can be applied to compare the performance by enhancing the PDT or CDSC task. Deep learning may also contribute to extract more relevant features and to perform classification. Furthermore, relevant features can be selected from various domains to contribute to document classification tasks using feature selection methods. Contextual information can be shared between the various layers of deep networks.

References

[1] Blitzer, J., Dredze, M., & Pereira, F. (June 2007). Biographies, Bollywood, boom-boxes, and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics.* 440-447.

[2] Bollegala, D., Weir, D., and Carroll, J. Aug (2013). Cross-Domain Sentiment Classification Using a Sentiment Sensitive Thesaurus. In IEEE Transactions on Knowledge and Data Engineering, vol. 25, No. 8. 1719-1731.

[3] Bollegala, D., Mu, T., and Goulermas. J. Y. Feb (2016). Cross-Domain Sentiment Classification Using Sentiment Sensitive Embeddings. In IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 2. 398-410.

[4] Duan, X., Zhou, Y., Jing, C., Zhang, L., and Chen, R. (2018). Cross-domain Sentiment Classification Based on Transfer Learning and Adversarial Network. IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China. 2302-2306.

[5] Heredia, B., Khoshgoftaar, T. M., Prusa J., and Crawford, M. (2016). Cross-Domain Sentiment Analysis: An Empirical Investigation. IEEE 17th International Conference on Information Reuse and Integration (IRI), Pittsburgh, PA. 160-165.

[6] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. ECML, 137–142.

[7] Li, Zheng, Wei, Ying, Zhang, Yu, and Yang, Qiang. (2018). Hierarchical Attention Transfer Network for Cross-Domain Sentiment Classification. In the Thirty-Second AAAI Conference on Artificial Intelligence, Hongkong.

[8] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability Vol. 1, No. 14. 281–297.

[9] Manshu, T., and Bing, W. (2019). Adding Prior Knowledge in Hierarchical Attention Neural Network for Cross Domain Sentiment Classification. In IEEE Access, vol. 7. 32578-32588.

[10] Manshu, T., and Xuemin, Z. (2019). CCHAN: An End to End Model for Cross Domain Sentiment Classification. In IEEE Access, vol. 7. 50232-50239.

[11] McCallum, Andrew, and Kamal Nigam. (1998). A comparison of event models for naive Bayes text classification. AAAI-98 Workshop on learning for text categorization. Vol. 752.

[12] Meng, Long, Yu, Zhao, and Liu. (2019). Cross-Domain Text Sentiment Analysis Based on CNN_FT Method. Information. Pages-162, volume10. May.

[13] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. (2013a). Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 3111–3119.

[14] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[15] Omurlu, Imran & Ture, Mevlut & Kurum, A.. (2008). Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease. Expert Systems with Applications. 34. 366-374. D.O.I- 10.1016/j.eswa.2006.09.004.

[16] Pan, Shinno Jialin, Xiaochuan Ni, Jian-JaoSun. Qiang Tang Zheng Chen. (2010). Cross-Domain Sentiment Classification via Spectral Feature Alignment. In Proceedings of the 19th international conference on World Wide Web. 751-760.

[17] Ponomareva, Natalia, and Thelwall, Mike. (2013). Semi-supervised vs. Cross-domain Graphs for Sentiment Analysis. Proceedings of the International Conference Recent Advances in Natural Language Processing. Sep. 571-578.

[18] Sammut, C., Webb, G.I. (Eds.). (2011). Encyclopedia of machine learning. Springer Science & Business Media.

[19] Tang, D., Qin, B., and Liu. T. (2016). Aspect level sentiment classification with deep memory network In the Conference on Empirical Methods on Natural Language Processing. 214–224.

[20] Wei, X., Lin, H., & LiangYang, W. (2017). Cross-domain Sentiment Classification via Constructing Semantic Correlation.

[21] Vapnik, Vladimir N. (1995). The nature of statistical learning theory. New York: Springer.

[22] Xiaojin Zhu and Zoubin Ghahramani. (2002). Learning from labeled and unlabeled data with label propagation.Technical report, Carnegie Mellon University.

[23] Yang, M., Yin, W., Qu, Q., Tu, W., Shen, Y., and Chen, X. (2019). Neural Attentive Network for Cross-Domain Aspect-level Sentiment Classification. In IEEE Transactions on Affective Computing.

[24] Zhang, B., Xu, X., Yang, M., Chen, X., and Ye, Y. (2018). Cross-Domain Sentiment Classification by Capsule Network with Semantic Rules. In IEEE Access, vol. 6. 58284-58294.

[25] Zhou, J., Huang J. X., Chen, Q., Hu, Q. V., Wang, T., and He, L. (2019). Deep Learning for Aspect-Level Sentiment Classification: Survey, Vision, and Challenges. In *IEEE Access*, vol. 7. 78454-78483.

[26] *Albon, Chris. (2018). Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep Learning (1st. ed.). O'Reilly Media, Inc.*

[27] Fangio Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. (2012). Cross-Domain Co-Extraction of Sentiment and Topic Lexicons. *ACL*

[28] Md Shad Akhtar, Ayush Kumar, Asif Ekbal, Pushpak Bhattacharyya. (2016). A Hybrid Deep Learning Architecture for Sentiment Analysis. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 482-493, Osaka, Japan, December.

[29] Wenyuan Dai, Qiang Yang, Guirong Xue, and Yong Yu. (2007). Boosting for transfer learning. In Proceedings of the 24th International Conference on Machine Learning, pages 193–200, Corvallis, Oregon, USA, June. ACM.

[30] Yaroslav Ganin, Hana Ajakan Hugo Larochelle, Francois Laviolette, Victor Lempitsky. (2016). Domain-Adversarial Training of Neural Networks. In Journal of Machine Learning Research 17, 1-35.

[31] Shinno Jialin Pan, Xiaochuan Ni, Jian-jaoSun. Qiang Tang, Zheng Chen. (2017). End-to-End Adversarial Memory Network for Cross-domain Sentiment. In Proceedings of the 19th international conference on World Wide Web, 751-760.