

# Cross Domain Adaptation using A Novel Convolution Neural Network

Ashwini Save<sup>1\*</sup> and Dr. Narendra Shekokar<sup>2</sup>

<sup>1</sup>PhD. Research Scholar, Comp. Engg. Dept., D. J. Sanghvi College of Engg, Mumbai, India.

<sup>2</sup>Professor, Comp. Engg. Dept., D. J. Sanghvi College of Engg, Mumbai, India.

ORCID: 0000-0001-7592-3932 (Ashwini), 0000-0002-2507-4140 (Dr. Narendra)

## Abstract:

Sentiment analysis has been widely researched in the recent times, especially with the advent of supervised machine learning and deep learning techniques. These supervised learning techniques require large amount of annotated data for working efficiently. But, this annotation of data for all the different products present in the market is an expensive prospect. Hence, a model or a system is required which trained on a particular domain should work in a cross domain environment as well.

The presence of many inherent challenges makes cross domain sentiment analysis highly inaccurate. The paper proposes a novel deep learning architecture for cross domain adaptation. After extensive experimentations with different parameters, like the number and placement of convolution layers and pooling layers, and hyper parameters, like the number and size of filters and that of windows, a multi-layered convolution neural network has been proposed which produces a comparatively more accurate cross domain adaptation results. The proposed system has been analysed and compared with state-of-the-art cross domain sentiment analysis systems and has shown to produce better results.

**Keywords:** Cross Domain Sentiment Analysis, Cross Domain Adaptation, Sentiment Analysis, Machine Learning, Deep Learning, Convolution Neural Network.

## 1. INTRODUCTION

With the increase in the global outreach of the World Wide Web more people are coming forward to express their opinions on people and things [1]. This rise of sharing opinions produces humongous amount of data in various domains [2, 3]. This huge data can be effectively used in the businesses if the sentiments are analyzed effectively. Sentiment Analysis is a field of study of Natural Language Processing and Machine learning where words expressed by any human is analyzed by the machine to determine the sentiment.

Generally classification algorithms are used for analyzing sentiments. Which means that data used for training has to be annotated. But, with the exponential increase in the number of products and product categories is becomes very difficult to annotate a large dataset in order to provide a respectable training for the classifiers. Considering the fact that the number of domains keeps on increasing it becomes very difficult, time consuming and costly to annotate the training data in each and

every possible domain. Due to these reasons cross domain adaptation approach is required which would be able to estimate and produce results even when the classifier is trained on different domain.

In cross domain sentiment analysis the classifier tries, as much as possible, to adapt itself to the unknown target domain. This adaptation of the classifier is the most important aspect of cross domain sentiment analysis. Consider the example of product reviews when training is given in one domain say on the 'books' but task is to classify user opinions from a different domain say 'electronics'. Here, when the classifier is trained on the books domain the features extracted by the classifier will be related to the story, author, etc., which is very different from what an opinion would be on the electronic products. For instance the word "novel" is fictional book and "novel" in electronics domain can be used as "new" idea of the product. But, even in such cases the classifier should be tuned in such a way that it is able to classify sentiments from the domain on which it wasn't trained; the classifier should adapt itself to the new domain.

Domain adaptation techniques are therefore important to leverage the source domain data and improve the accuracy of the system while testing the target domain. It is important to reduce distribution gap between training dataset and testing dataset. Hence, many techniques have been put forth to achieve cross domain sentiment analysis in the recent times. The paper analysis different domain adaptation techniques and proposes a new deep learning technique for cross domain adaptation.

## 2. REVIEW OF RELATED LITERATURE

In this section review of some important techniques which have proposed for cross domain sentiment analysis has been carried out.

One of the most important technique in cross domain adaptation is the use of pivots by John Blitzer, et. al. [4], where they propose a method for selection of pivots. In this, first the frequency of the words are found out from the training data, the words with maximum frequency can be called as the pivot of the source domain. Further to increase the accuracy of the domain adaptation paper [5] proposes to provide the classifier with a labelled data. Here the pivots are selected based on two criteria, the words with the highest frequency and their mutual information with the labelled source data.

Danushka Bollegala, et. al. [6] on the other hand skips the whole process of pivot selection and assumes that pivots are already given. This paper proposes three objective function which tries to optimize to get better domain adaptation.

Another Solution to domain adaptation is joint topic modelling approach proposed by Zhen Hai, et. al. [7]. In this supervised learning approach, with the help of labelled source data aspects terms are generated for the target domain. With the help of these aspect terms classifier determines the domain adaptation. Shenghua Liu, et. al. [8] proposes a transduction method to identify the topic or the domain of the target data to further work on sentiment analysis on twitter data. In this global words and non-global words are taken into consideration. Polarity of all the global words are calculated. Non-global words are used to find the domain. Kun-Hu Luo, et. al. [9] proposes a simple method based on emotional words for achieving of cross domain sentiment analysis.

Yuewei Lin, et. al. [10] proposed method of developing joint subspaces. This technique is based on relation between source-target subspaces and relation within anchor subspaces. The major drawback of this system is the source and target domains pairs will have to be identified beforehand. Shuang Li, et. al. [11] proposed a technique to train the classifier after reweighting of the data points. In this with the help of the reweighted source data point the reweighted target data point is predicted. And with the help of the predicted target data point the unknown data point is predicted by using label propagation. Deep learning approach has been used by Debora Nozza, et. al. [12] to give the solution to domain adaptation problem. Using deep learning concept the feature values of the source domain is transformed. And after transformation if the feature values are similar to the target domain features values then it can be said that the corresponding features are similar as well. Whereas, Wenjie Zhang, et. al. [13] proposes a system based on similar words and distinct words. It find outs the similarity between similar words and then features of the source domain can be approximated to the feature of the target domain. Danushka Bollegala, et. al. [14] propose to develop and use a sentiment sensitive thesaurus for sentiment classification. This system is complex as it uses pointwise mutual information very extensively. Also, for the preparation of an exhaustive thesaurus a large number of labelled training data will be required which is detrimental and expensive. Min Xiao and Yuhong Guo [15] has given the solution for domain adaptation with a feature space independent semi-supervised kernel matching method. In this system mapping of the target data points similar to the source data points is done by matching the target kernel matrix to a submatrix of the source kernel matrix based on a Hilbert Schmidt Independence Criterion. Independence Criterion. Fangzhao Wu, et. al. [16] propose a collaborative learning system for sentiment classification to deal with sparse annotated data of few domains. The system also encourages sharing of sentiment information between similar domains. This is done by using the similarities between domains. Also, two kinds of domain similarity measures have been implemented in the system, one based on textual similarity and the other one based on sentiment expressions.

Even though there has been many research in the recent past there is ample scope of improvement in order to get better adaptation accuracy. As the domain accuracy produced till now have not been at a level at it is desired, this area has been part of the research area than being used in the industry extensively. Also, with the achievements of Deep Learning techniques in other fields it is imperative that it advantages should be utilized in this field of research as well.

Due to the scarcity of enough annotated data for proper training of the classifier, a classifier trained on one domain has to perform respectfully for the prediction of data from another domain. This process of predicting the results of one domain with the help of the classifier trained on another domain is cross domain sentiment analysis.

In cross domain sentiment analysis, domain adaptation is the most important factor that affects proper analysis. So, an approach is required which would be able to adapt properly to the target domain would be helpful in increasing the sentiment analysis accuracy while dealing with multiple domain.

### 3. PROPOSED SYSTEM

Deep learning techniques have provided important breakthroughs in many different fields of research. Leveraging the advantages provided by deep learning in the field of cross domain sentiment analysis the proposed system makes use of an important deep learning technique called the Convolution Neural Network. Convolution Neural Networks have been extensively used and researched for working with images. In recent times their effectiveness have also been showcased for analyzing texts as well. Taking this concept further the paper proposes a convolution neural network architecture for cross domain sentiment analysis.

A CNN usually consists of three type of layers, Convolution layer, Pooling layer and Fully connected layer [17]. The convolution layer [18] extracts features from the data, the job of the pooling layer is to down size the data so that the computing complexity reduces [19] and the fully connected layer Artificial Neural Network is used for the classification purpose. These layers can be used in any combination depending upon the data and the research.

There are also many other parameters and hyper parameters which needs to be considered while dealing with convolution neural network research. The research parameters include the number of convolution layers to be used, the number of pooling layers to be used, and the order in which these two are applied needs to be found out as well. Apart from the parameters there a certain hyper parameters which needs research as well, these include the number filters to be applied on the convolutions layers and their size, the window size of pooling layer has to be determined as well. Apart from these, there are other aspects which needs research as well like the type activations functions to be used and the probability distributions functions to be applied. So, the research in convolution neural network requires a multifaceted approach in order to identify a costumed architecture for a specified problem.

This section has been subdivided into 2 parts. In the first part the dataset used for the training and testing purpose has been discussed, and the second part explains the experimental setup of the proposed system.

### 3.1 Dataset:

Most of the research which have been carried out in the field of cross domain adaptation have used the Amazon Review Dataset [20] for testing the accuracy of the system proposed.

The Amazon review dataset have been divided into four domains or categories. These review categories are reviews for Book, for Digital Video Disks, Kitchen appliances and Electronics. For the purpose of sentiment analysis each of these product categories have one thousand positive and one thousand negatives reviews.

The proposed system also makes use of this Amazon review dataset to prove the effectiveness of the system and to prove that the proposed system's accuracy compared to the baseline accuracy parameters.

### 3.2 System Architecture:

The proposed system makes use of different number of convolution, pooling, and dense layers.

Convolution is basically a function derived from the combination of two different functions [21]. Convolution can be given as:

$$f(x) * g(x) = f'(x)$$

Where,

$f(x)$ : input matrix.

$g(x)$ : filter with size less than the size of  $f(x)$ .

\*: refers to the multiplication and averaging of the values in  $f(x)$  mapped under the filter and weights of  $g(x)$ .

$f'(x)$ : resulting matrix, also called as feature map.

Generally, the size of the resulting matrix gets reduced after applying convolution. But, for better understanding and in order to apply multiple layers the resulting matrix is padded to make it of the same size as the input matrix. This can be given as [22]:

$$n_{out} = \left\lceil \frac{n_{in} + 2p - k}{s} \right\rceil + 1$$

Where,

$n_{in}$ : number of input features

$n_{out}$ : number of output features

$p$ : convolution padding size

$k$ : convolution kernel size

$s$ : convolution stride size

In the proposed system Rectified Linear Unit (ReLU) has been used as an activation function for all the convolution layers. Rectified Linear Unit is a liner function which produces an output of zero if the input is either zero or a negative number, and if the input is a positive number than this value itself is given as an output [23]. ReLU can be given as:

$$f(x) = \max(x, 0)$$

There are three types of pooling methods, Average Pooling, Minimum Pooling and Maximum Pooling. In order to extract prominent features Maximum Pooling or Max Pooling has been used in the proposed system. The Max-Pooling can be given as [24]:

$$y = \max_{i,j=1}^{h,w} x_{i,j}$$

Here,

$h, w$ : maximum size (height and width) of the Max Pooling window.

$x_{i,j}$ : input matrix

That is, the maximum value within the max pooling window will be the output of this layer.

Using these concepts of convolution neural network the system has been designed. In the proposed system before applying any deep learning layers the review text ( $r_1$ ) has to be converted into vectors ( $e_{r_1}$ ). This process of converting text words into word vectors is called as word embedding. In the proposed system Word2Vec technique has been used for word embedding.

Before any words are replaced with its vector equivalent the Word2Vec has to be trained on different words to find out the relation between these words. These relations decide the vector value.

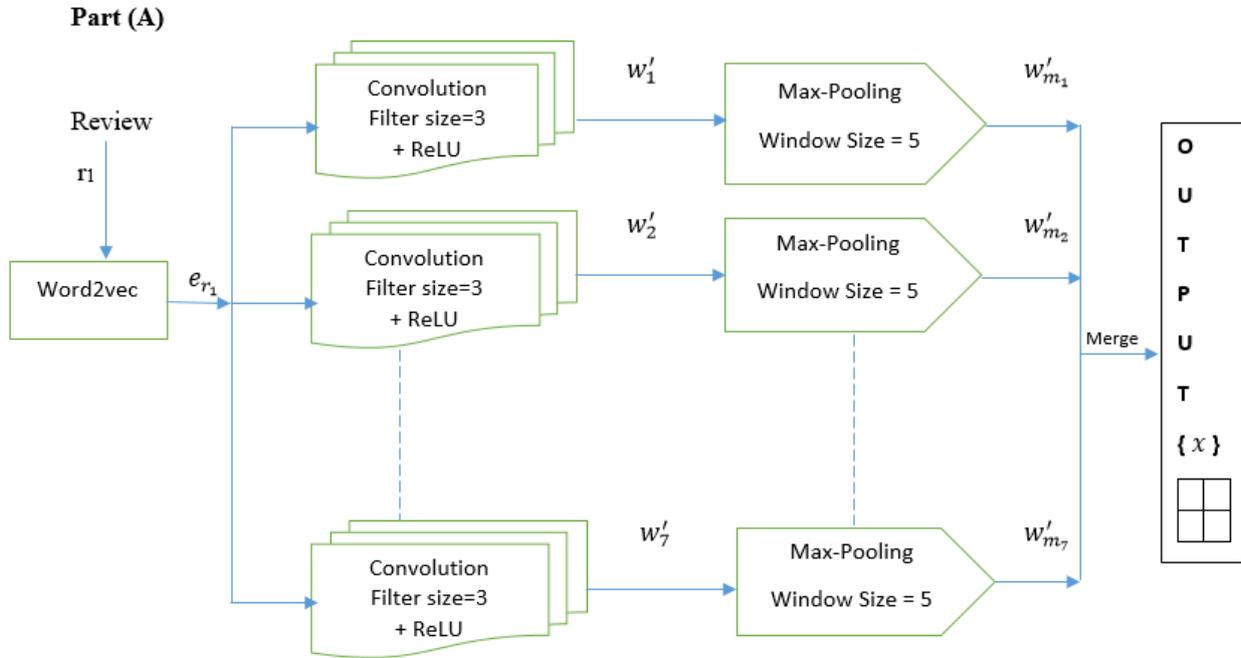
There are essentially two ways in which Word2Vec can be used, either a pre-trained model can be used or training the Word2Vec system on the available datasets which is going to be used for a specified purpose. These pre-trained models are trained on large datasets, like the Wikipedia and are available on the internet. But, from experimentations it was found out that rather than a pre-trained model, a model generated by training it on the available dataset were given better results. So, in the proposed system a Word2Vec model was created using the twelve thousand available reviews and this model generated around forty thousand vectors for the corresponding unique words.

Once the word embedding has been completed and the review text has been converted into their equivalent word vectors the convolution neural network is applied.

As shown in the Figure 1, seven convolution layers with seven different filter sizes (filter sizes from 3 to 9) are applied on the word vectors. Here, Rectified Linear Unit (ReLU) has been used as an activation function to convert all negative number into zero. These seven convolution layers are applied independent of each other. The input to these convolution layers is a review  $e_{r_1}$ , the output of these layers are  $w'_1, w'_2, \dots$

-  $w'_7$ . These given as an input to the Max Pooling layer with window size 5, producing outputs  $w'_{m_1}, w'_{m_2}, \dots, w'_{m_7}$ .

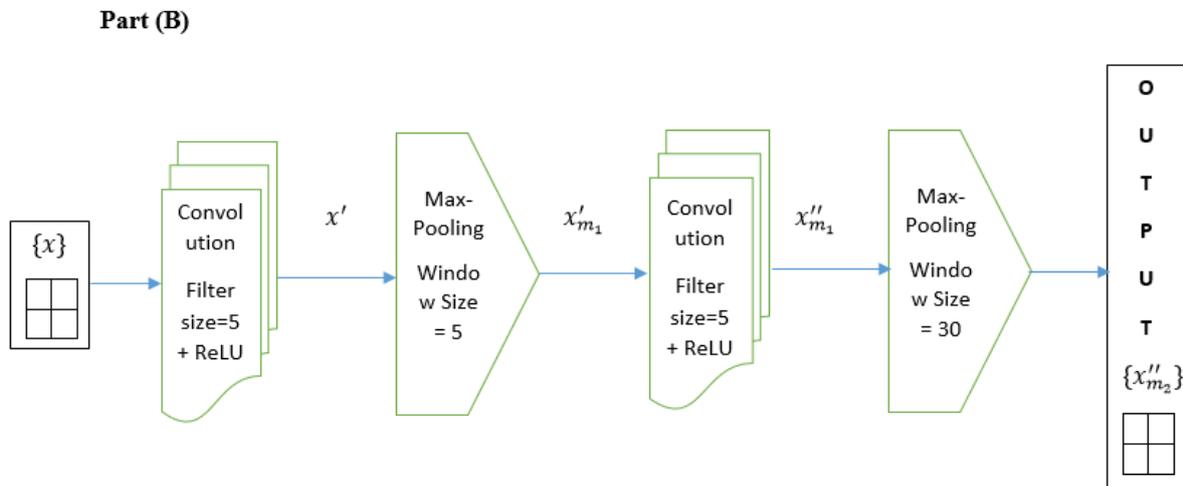
These seven outputs ( $w'_{m_1}, w'_{m_2}, \dots, w'_{m_7}$ ) of the Max Pooling are concatenated to get a single output  $x$ . i.e.  $x = w'_{m_1} + w'_{m_2} + w'_{m_3} + w'_{m_4} + w'_{m_5} + w'_{m_6} + w'_{m_7}$



**Fig 1: CNN Architecture for Domain Adaptation (Part A)**

As it can be seen in Figure 2, Convolution layer is applied with Filter size 5 and ReLU as an activation function on  $x$ , producing an output  $x'$ . Max Pooling layer with window size 5 has been applied on  $x'$ , producing an output  $x'_{m_1}$ .

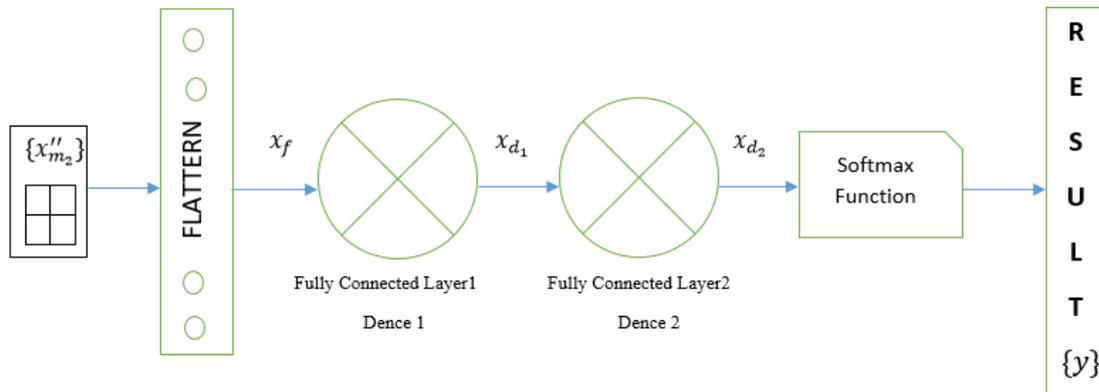
A convolution layer with filter size 5 and ReLU activation function has been applied on  $x'_{m_1}$ , obtain the output  $x''_{m_1}$ . Max Pooling layer with the window of size 30 has been applied on this data, thus producing  $x''_{m_2}$ .



**Fig 2: CNN Architecture for Domain Adaptation (Part B)**

As shown in Figure 3,  $x''_{m_2}$ , which is in a matrix form is then flattened to get  $x_f$ , which is a vector form. The flattening is performed so that Artificial Neural Network can be applied.

**Part C:**



**Fig 3: CNN Architecture for Domain Adaptation (Part C)**

Two Artificial Neural Networks, also called as dense network or the fully connected network, have been applied one after the other. The output of the first fully connected network ( $x_{d_1}$ ) contains 128 neurons and the output of the second dense network ( $x_{d_2}$ ) contains 2 neurons. The 2 neurons here indicate the 2 classes in which classification is to be performed.

In order to get the classes the values at these 2 neurons has to be converted into probability, this probability score is the final output ( $y$ ). Softmax probability distribution function has been used to convert the values into probability. With the help of this probability score the classes are assigned. For getting results into probability distribution, two alternatives of softmax functions were taken into consideration. Regular Softmax Function and Taylors Softmax Function.

Regular Softmax function converts the given  $k$  real number vectors into a probability distribution containing  $k$  probabilities [25]. It is given as [26]:

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

Here, the softmax function calculates the exponent of each input value and then divides it by the sum of all the input exponents.

Whereas Taylors Softmax Function is given as,

$$f_{tay\_soft}(O)_k = \frac{1 + O_k + \frac{1}{2}O_k^2}{\sum_{i=1}^D(1 + O_i + \frac{1}{2}O_i^2)}$$

To elaborate how Regular Softmax function is having better functioning, the following example is given. In neural network to make prediction where the thing-to-predict can be one of three possible values. A neural network classifier would accept numeric inputs and then generate a preliminary output of three values say (1.55, 2.30, 0.90). but then normalize the preliminary outputs so that they sum to 1.0 and can be interpreted as probabilities.

Input Vector =  $X_i = [1.55 \quad 2.30 \quad 0.90]$

Using Regular Softmax Function:

$\exp(1.55) = 4.71$

$\exp(2.30) = 9.97$

$\exp(0.90) = 2.46$

sum = 17.15

$\text{softmax}(1.55) = 4.71 / 17.15 = 0.28$

$\text{softmax}(2.30) = 9.97 / 17.15 = \mathbf{0.58}$

$\text{softmax}(0.90) = 2.46 / 17.15 = 0.14$

Sum=1

Using Taylors Softmax Function:

$\text{taylor}(1.55) = 1.0 + 1.55 + 0.5 * (1.55)^2 = 3.75$

$\text{taylor}(2.30) = 1.0 + 2.30 + 0.5 * (2.30)^2 = 5.96$

$\text{taylor}(0.90) = 1.0 + 0.90 + 0.5 * (0.90)^2 = 2.31$

sum = 12.00

$\text{taylor-soft}(1.55) = 3.75 / 12.00 = 0.31$

$\text{taylor-soft}(2.30) = 5.96 / 12.00 = \mathbf{0.50}$

$\text{taylor-soft}(0.90) = 2.31 / 12.00 = 0.19$

Sum=1

From above mathematical discussion it gives clear understanding of the both alternatives of softmax function. The Regular softmax function is giving probability distribution at higher side than the Taylors softmax function. So Regular softmax function can be consider as best giving results to use in the architecture.

For same domain training and testing 80:20 rule has been used. That is, to test the accuracy of the model trained on Book domain and tested on Book domain itself 80% of the data has been used for training and the remaining 20% for testing.

This is the whole experimental setup which has been used in order to attain high cross domain adaptation accuracy.

#### 4. RESULTS AND ANALYSIS:

In this section the results for cross domain adaptation obtained by the proposed system has been compared with some of the state of the art cross domain adaptation papers, which have been reviewed in the previous sections.

Table 1 shows the domain adaptation accuracy of the proposed CNN model.

**Table 1:** Adaptation accuracy: Proposed System

		Target Domain			
		BOOK	DVD	ELECTRONICS	KITCHEN
Source Domain	BOOK	88.71	86.56	88	96.73
	DVD	87.63	85.79	87.25	96.73
	ELECTRONICS	84.95	83.46	84.5	98.7
	KITCHEN	85.48	84.75	86.75	92.71

In table 1, the source domain means the domains on which the CNN cross domain adaptation model, proposed in this paper, has been trained on and the target domains signifies the domain on which the trained model has been tested upon.

Table 2 shows the detailed comparison of the proposed system with other cross domain sentiment analysis techniques.

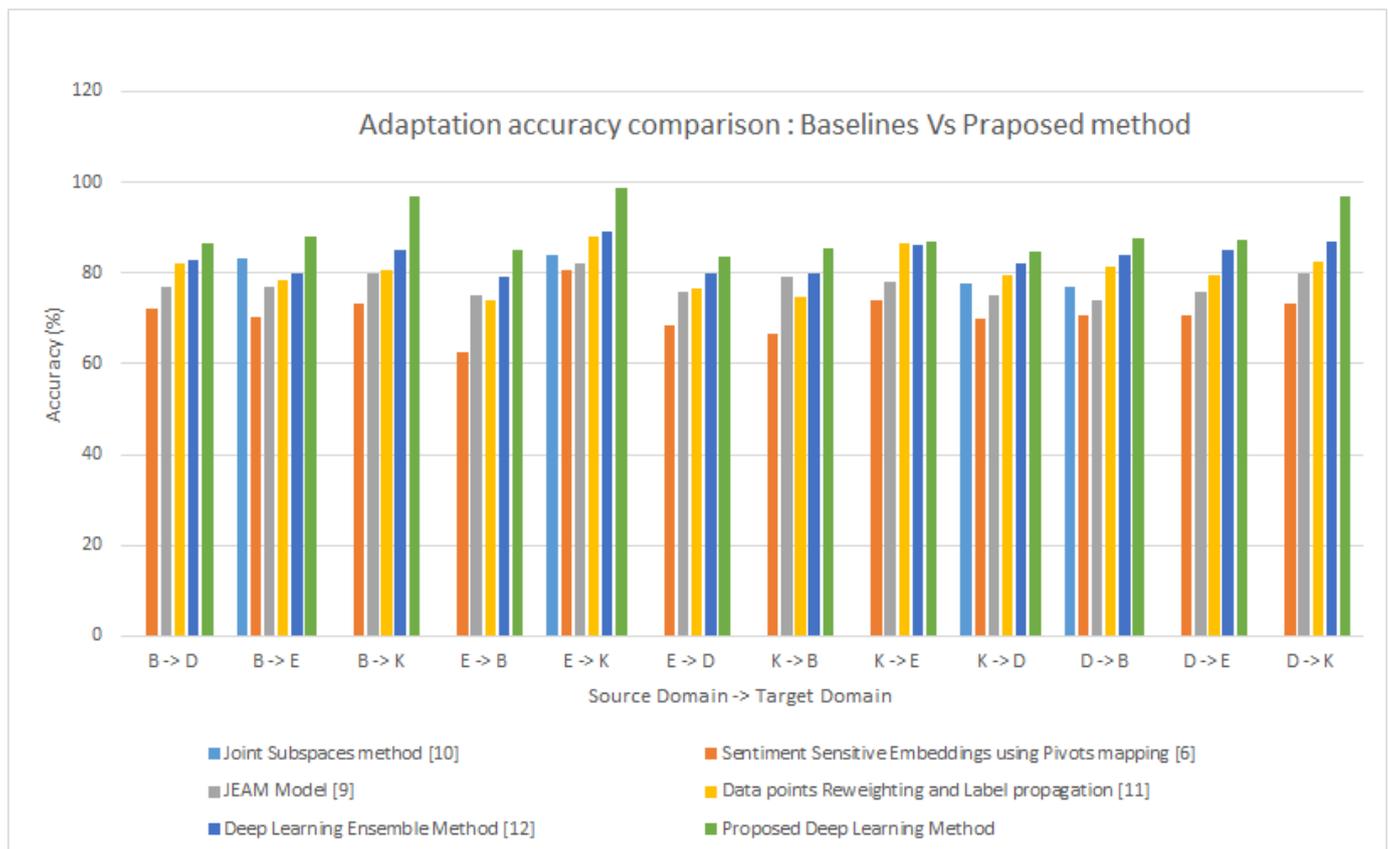
**Table 2:** Adaptation accuracy comparison: Baselines Vs Proposed method

Source Domain	Target Domain	Joint Subspaces method [10]	Sentiment Sensitive Embeddings using Pivots mapping [6]	JEAM Model [9]	Data points Reweighting and Label propagation [11]	Deep Learning Ensemble Method [12]	Proposed Deep Learning Method
B	D	--	72.12	77	82.0	83	<b>86.56</b>
B	E	83.2	70.27	77	78.3	80	<b>88</b>
B	K	--	73.36	80	80.6	85	<b>96.73</b>
E	B	--	62.46	75	74.0	79	<b>84.95</b>
E	K	84.1	80.65	82	87.9	89	<b>98.7</b>
E	D	--	68.28	76	76.6	80	<b>83.46</b>
K	B	--	66.49	79	74.8	80	<b>85.48</b>
K	E	--	74.05	78	86.6	86	<b>86.75</b>
K	D	77.8	70.07	75	79.4	82	<b>84.75</b>
D	B	77.0	70.52	74	81.5	84	<b>87.63</b>
D	E	--	70.52	76	79.5	85	<b>87.25</b>
D	K	--	73.11	80	82.5	87	<b>96.73</b>

From the Table 2 it can be seen that the proposed model produces better results than the compared techniques. Also, it can be seen that for all the techniques the best result has been produced for system trained on Electronics and tested on Kitchen data (E -> K). The highest accuracy percentage produced by the compared techniques are, by paper [10] is 84.1%, by paper [6] is 80.65%, by paper [9] is 82%, by paper [11] is 87.9%, by the paper [12] is 89% and by the proposed system is 98.7%, for E -> K. The lowest results produced by

paper [10] is 77% for D -> B, by paper [6] is 62.46% for E -> B, by paper [9] is 74% for D -> B, by paper [11] is 74% for E -> B, by paper [12] is 79% E -> B and the proposed system it is 83.46% for E -> D.

Figure 4 gives the bar chart comparison of the proposed model's accuracy results compared with the other baseline model's accuracy results.



**Fig 4:** Adaptation accuracy comparison: Baselines Vs Proposed method

From the Figure 4 it can be clearly seen that the proposed CNN architecture model gives better results in all the cases of cross domain adaptation when compared with the baseline models.

On an average the proposed model is producing an accuracy of around 88%. The highest accuracy result of 96.73 % has been achieved for cross domain adaptation for the model trained on Book domain and tested on Kitchen domain and also when the model is trained on DVD domain and tested on Kitchen domain.

From the results obtained, which have been given in Table 1, Table 2 and Figure 4 it can be clearly seen that the proposed model comparatively produces better results in all the possible cross domain combinations. This shows that the proposed model has worked satisfactorily in all possible combinations of training and testing datasets from the Amazon Review dataset.

## 5. CONCLUSION

Cross Domain Adaptation has become an important area of research in the recent times especially with the increase in the number of products and product categories. This increase in the products and product categories has demanded the availability of labelled trained data for each, which is an expensive proposition. So, cross domain adaptation is an important approach which enables an effective transition. So, this paper proposes a new technique in the field of Cross Domain Adaptation by given a model which uses Convolution Neural Network to increase the accuracy of domain adaptation.

The paper proposes a unique network of Convolution and Pooling layers with activation functions which provides for better learning of the given data, which in turn helps in producing better results for domain adaptation.

Also, the results prove that, compared to the baseline cross domain adaptation algorithms the proposed system produces far better results. With the highest accuracy reaching more than 95%, the paper gives a technique for cross domain adaptation which is better than many state of the art cross domain adaptation techniques.

## REFERENCE

- [1] Editorial Staff. (2017, Jan 12). Internet Trends, Stats & Facts in the U.S. and Worldwide 2017 [Online]. Available: [http://www.hotel-online.com/press\\_releases/release\\_/internet-trends-stats-facts-in-the-u.s.-and-worldwide-2017](http://www.hotel-online.com/press_releases/release_/internet-trends-stats-facts-in-the-u.s.-and-worldwide-2017), Last accessed 19<sup>th</sup> Feb, 2017.
- [2] Editorial Staff. (2016, June 8). How Internet in India Will Look in 2020: 12 Exciting Statistics! [Online]. Available: <http://trak.in/tags/business/2016/06/08/india-internet-growth-2020-statistics/>, Last accessed on 19<sup>th</sup> Feb, 2017.
- [3] Neeraj M. (2016, Feb. 8). Mobile Internet Users In India 2016: 371 Mn by June, 76% Growth In 2015 [Online]. Available: <https://dazeinfo.com/2016/02/08/mobile-internet-users-in-india-2016-smartphone-adoption-2015/>, Last Accessed on 19<sup>th</sup> Feb 2017.
- [4] J. Blitzer, M. Dredze and F. Pereira, "Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification", ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, June 23-30, 2007.
- [5] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning", Empirical Methods in Natural Language Processing (EMNLP), 2006.
- [6] D. Bollegala, T. Mu and J. Y. Goulermas, "Cross-domain Sentiment Classification using Sentiment Sensitive Embeddings", IEEE Transactions on Knowledge and Data Engineering, 2015.
- [7] Z. Hai, G. Cong, K. Chang, P. Cheng, and C. Miao, "Analyzing Sentiments in One Go: A Supervised Joint Topic Modeling Approach", IEEE Transactions on Knowledge and Data Engineering, 2017.
- [8] S. Liu, X. Cheng, F. Li, and F. Li, "TASC: Topic-Adaptive Sentiment Classification on Dynamic Tweets", IEEE Transactions on Knowledge and Data Engineering, 2014.
- [9] K. Luo, Z. Deng, L. Wei, and H. Yu, "JEAM: A Novel Model for Cross-Domain Sentiment Classification Based on Emotion Analysis", Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17-21 September 2015, Association for Computational Linguistics, pages 2503–2508.
- [10] Y. Lin, J. Chen, Y. Cao, Y. Zhou, L. Zhang, Y. Y. Tang, and S. Wang, "Cross-Domain Recognition by Identifying Joint Subspaces of Source Domain and Target Domain", IEEE Transactions On Cybernetics, 2016.
- [11] S. Li, S. Song, and G. Huang, "Prediction Reweighting for Domain Adaptation", IEEE Transactions On Neural Networks And Learning Systems, 2016.
- [12] D. Nozza, E. Fersini, and E. Messina, "Deep Learning and Ensemble Methods for Domain Adaptation", IEEE 28th International Conference on Tools with Artificial Intelligence, 2016, pp. 184-189.
- [13] W. Zhang, H. Zhang, D. Wang, R. Liu, H. Zhang, X. Jiang, and Y. Chen, "Transfer Learning by Linking Similar Feature Clusters for Sentiment Classification", IEEE 28th International Conference on Tools with Artificial Intelligence, 2016, pp. 1019-1026.
- [14] D. Bollegala, D. Weir and J. Carroll, "Cross-Domain Sentiment Classification using a Sentiment Sensitive Thesaurus", IEEE Transactions On Knowledge And Data Engineering.
- [15] M. Xiao and Y. Guo, "Feature Space Independent Semi-Supervised Domain Adaptation via Kernel Matching", IEEE Transactions On Pattern Analysis And Machine Intelligence, 2015.
- [16] F. Wu, Z. Yuan, and Y. Huang, "Collaboratively Training Sentiment Classifiers for Multiple Domains", IEEE Transactions on Knowledge and Data Engineering, 2017.
- [17] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks", [https://www.researchgate.net/publication/285164623\\_An\\_Introduction\\_to\\_Convolutional\\_Neural\\_Networks](https://www.researchgate.net/publication/285164623_An_Introduction_to_Convolutional_Neural_Networks), Last Accessed on: 23<sup>rd</sup> Jan, 2019
- [18] I. Goodfellow, Y. Bengio and A. Courville, "Deep Learning", MIT Press, 2016, pp. 327-328.
- [19] H. Wu and X. Gu, "Max-Pooling Dropout for Regularization of Convolutional Neural Networks", <https://arxiv.org/ftp/arxiv/papers/1512/1512.01400.pdf>, Last Accessed on: 27<sup>th</sup> Jan, 2019.
- [20] <https://www.cs.jhu.edu/~mdredze/datasets/sentiment/>, Last Accessed on: 27<sup>th</sup> Jan, 2019.
- [21] D. Cornelisse, "An intuitive guide to Convolutional Neural Networks", <https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/>, Last Accessed on: 25<sup>th</sup> Jan 2020.
- [22] D. Ha The Hien, "A guide to receptive field arithmetic for Convolutional Neural Networks", <https://medium.com/mlreview/a-guide-to-receptive-field-arithmetic-for-convolutional-neural-networks-e0f514068807>, Last Accessed on: 25<sup>th</sup> Jan, 2020.
- [23] J. Brownlee, "A Gentle Introduction to the Rectified Linear Unit (ReLU)", <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>, Last Accessed on: 25<sup>th</sup> Jan, 2020.

- [24] D. Mishkin, “CNNs: From the basics to the recent advancements”, <https://www.slideshare.net/ducha/cnns-from-the-basics-to-recent-advances>, Last Assessed on: 25<sup>th</sup> Jan, 2020.
- [25] Editors, “Softmax Function”, [https://en.wikipedia.org/wiki/Softmax\\_function](https://en.wikipedia.org/wiki/Softmax_function), Last Accessed on: 25<sup>th</sup> Jan, 2020.
- [26] Uniqtech, “Understand Softmax Function in minutes”, <https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d>, Last Accessed on: 25<sup>th</sup> Jan, 2020.
- [27] A. de Brébisson, and P. Vincent, “An Exploration Of Softmax Alternatives Belonging To The Spherical Loss Family”, ICLR 2016.