INDIAN INSTITUTE OF
INFORMATION TECHNOLOGY, ALLAHABAD

Cross-Domain Sentiment Analysis via Spectral Feature Alignment

Ayush Agnihotri	IIM2015004
Nidheesh Pandey	IIM2015501
Abhishek Pasi	ICM2015002
Shreyansh Gupta	IIM2015001
Vishal Kumar Singh	IIT2015141

UNDER SUPERVISION
of
DR. K P SINGH

DECLARATION

We hereby declare that the work presented in this project report entitled " ***CROSS DOMAIN SENTIMENT ANALYSIS USING TRANSFER LEARNING (via Spectral Feature Alignment)*** ", submitted as End-Semester Report of 6th Semester B.Tech.(IT) at **Indian Institute of Information Technology, Allahabad**, is an authenticated record of our original work carried out from January 2018 to May 2018 under the guidance of **Dr. K P Singh**. Due acknowledgements have been made in the text to all other material used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Signature: _____

Date: _____

Contents

List of Figures

List of Tables

Cross-Domain Sentiment Analysis via Spectral Feature Alignment

Abstract

The main aim of this report is to define an approach to implement cross-domain sentiment analysis. Although traditional classification algorithms can be used to train sentiment classifiers from manually labeled text data but the labeling work can be time-consuming and expensive. Moreover,If we directly apply a classifier trained in one domain to other domains, the performance will be very low due to the differences between these domains.Therefore we are using a Transfer Learning approach for classification of Cross Domain datasets.

In our approach, we took datasets from two different but related domains,and used one of them as source and other as target domain dataset. We extract the features from both of these datasets and classify the extracted features into domain independent features (pivot features) and domain specific features. We used bipartite graph representation between domain independent and domain specific features to find co-ocurrence between these two types of features.Spectral Clustering algorithm is then applied to cluster the domain specific features around closely related pivot features. This reduces the gap between domain-specific words of the two domains.Now we have a new augmented set of features containing all the features of source domain + the newly found clustered features. This new feature representation is then used to train the sentiment classifier, to classify the target domain accurately.

Since we have labeled dataset in target domain, so we can test the accuracy of our model by direct comparison. Comparison with other cross domain sentiment analysis algorithms and various optimisations techniques to improve the accuracy of the model will be added in future version of this report

Introduction

The world is getting flooded by textual data everyday. It is very important to extract infromation from this data for better understanding of the world. These data includes product reviews, blogs, public opinions, stock trends etc. The informations hidden in these data is used by bussiness firms, govt. agencies to formulate the policies and future plans. Inorder to extract this important inforamtion we need labeled datasets. Most of the textual data present today is unlabeled which makes it highly difficult to analyze them.

Earlier works in sentiment analysis involved various supervised learning algortihms. For using supervised learning methodologies we require labeled datasets. Labeling a dataset manually is time consuming and expensive. Also the earlier models were domain specific (i.e. when a classifier trained in one domain is applied in other domains, then it performs poorly). So, there does not exist a general classifier which can be used for sentiment analysis in multiple domains.

In this report we would like to present a model which will do Sentiment Analysis on various domains and for unlabeled datasets. It does this by learning the knowledge from a labeled dataset and transfering the learnt knowledge. We took a labeled dataset as source domain inorder to train our classifier for target domain.

The main algorithm we are implementing is Spectral Feature Alignment (SFA) algorithm which helped us to represent cross domain data in a combined way to form as a new set of features. SFA does this by clustering domain specific features around domain independent features depending upon the co-ocurrence between the domain specific and domain independent features. These features are represented uses bipartitte graph.

The concept is that,

1. if two domain-specific features are connected to many common domain-independent features, then they tend to be very related and will be aligned to a same cluster with high probability,
2. if two domain-independent features are connected to many common domain-specific features, then they tend to be very related and will be aligned to a

same cluster with high probability.

We have implemented this using Spectral Clustering (based on graph spectral theory) on the bi-partite graph to co-align domain-specific and domain-independent words into a set of feature-clusters. The clusters helps us to reduce the degree of mismatch between domain specific features of both domains. So, our final feature set is based on this combined cluster representation. Then our classsfier is trained on this new feature set which is able to do cross domain sentiment analysis with greater accuracy level than traditional methods.

We'll optimise the accuracy of our classifier model using SFA and do the corresponding experiments .

Literature survey

a). **Jialin** et. al.[1] focuses on how to do sentiment analysis on cross domains using SFA algorithm. Here first domain-specific & domain-independent features are extracted & unified into clusters using Spectral Feature Clustering. This feature set is then used by classifier for cross-domain sentiment analysis.

b). **Lin** et. al.[2] proposed a general ensemble technique which takes into account model application, model weight & strategies for selecting most related models with respect to a target node. Here two strategies, cosine function and taxonomy-based regression model (TBRM) are proposed to select most related models.

c). **WuFangzhao** et. al.[3] proposes a novel approach to train domain-specific sentiment classifiers by fusing the sentiment knowledge from multiple sources. The 1^{st} source is sentiment lexicons , 2^{nd} is sentiment classifier of multiple domain, 3^{rd} unlabeled data in target domain, 4^{th} is labeled data in target domain. It alsoproposes a unified framework to fuse these four kinds of & train domain-specific sentiment classifier for target domain.

d). **Nelakurthi** et. al.[4] addresses this problem by explicitly modeling the human factor related to sentiment classification. To this end, he proposed a new graph-based approach named U-Cross, which models the relatedness of different domains via both the shared users and keywords. It is non-parametric and semi-supervised in nature.

e). **Fangzhao Wu** et. al.[5] gave a new sentiment domain adaptation approach by adapting the sentiment knowledge in general-purpose sentiment lexicons to a specific domain is proposed. Moreover it also proposes a unified framework to incorporate these different kinds of sentiment knowledge and learn an accurate domain-specific sentiment classifier for target domain.

Problem Statement

Problem Definition :

Given two domains D_{src} and D_{tar} , where D_{src} and D_{tar} are referred to as a source domain and a target domain respectively, suppose we have a set of labeled training sentiment data $\mathscr{D}_{src} = \{(x_{srci} , y_{srci})\}_{i=1}^{n_{src}}$ in D_{src} , and some unlabeled sentiment data tar , $\mathscr{D}_{tar} = \{x_{tarj}\}_{j=1}^{n_{tar}}$ in D_{tar} . The task of cross-domains sentiment classification is to learn an accurate classifier to predict the polarity of unseen sentiment data from D_{tar} .

Some definitions :

Definition 1:

Domain :-A domain D denotes a semantic concept.
Example :- different types of products like toys, electronics, books, kitchen can be regarded as domains.

Definition 2:

Sentiment :- In Context of a given domain D, Sentiment data is a text document which conveys some opinion.

Methodology Flow Diagram

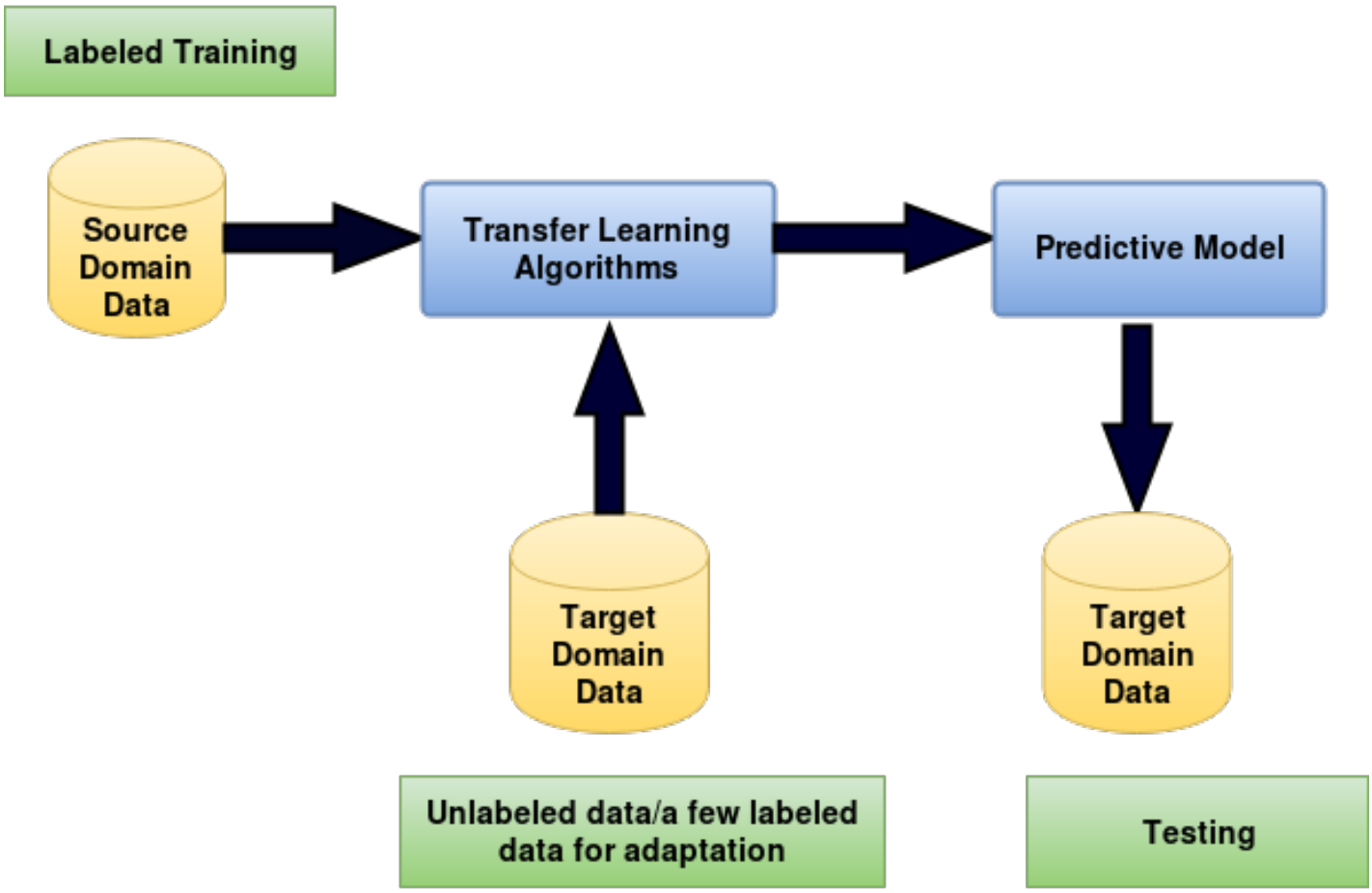

Fig - 4.1: *Diagram representing Flow diagram of methodology*

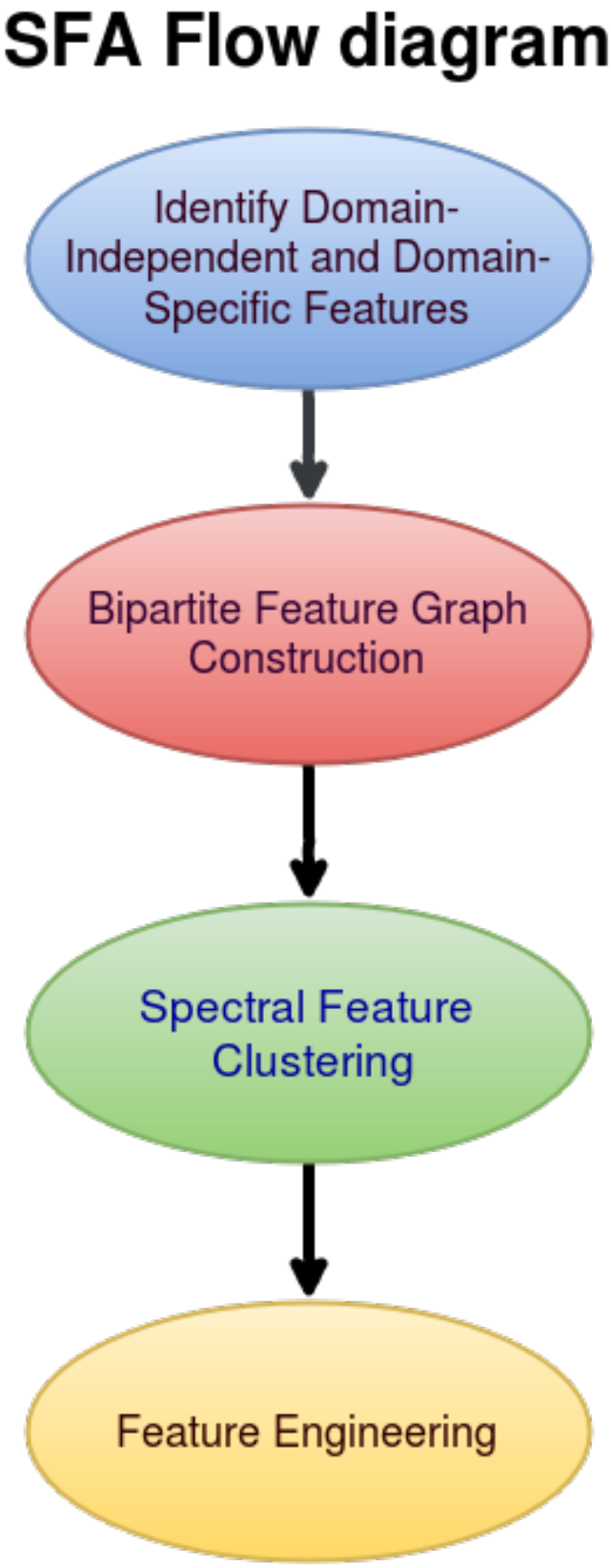

Fig - 4.2: Figure representing flow of SFA Algorithm

Methodology

The main challenge of cross domain sentiment analysis (transfer learning) is to find representation of cross domain sentiment data such that the gap between the source domain and target domain can be reduced.

We are using a Transfer Learning Algorithm called Spectral Feature Alignment (SFA) algorithm to find a new representation for cross-domain sentiment data, such that the gap between domains can be reduced. SFA uses some domain-independent words as a bridge to construct a bipartite graph to model the co-occurrence relationship between domain-specific words and domain-independent words.

This method involves following tasks -

1) To Recognize domain independent and domain specific features.
2) To align domain-specific features from different domains for cross-domain sentiment classification using spectral clustering method.
3) Train classifiers on the source using augmented features (original features + new features).

Recognising Domain-specific and Domain-independent Features

Recognising Domain Independent and Domain Specific features is an important step in cross domain sentiment analysis. This is so because users express same sentiment using different words in different domains. Consider a case of domain specific words :- Example - 'sharp' , 'compact' are used to express positive sentiment and 'blurry' for negative sentiment in case of Electronics while the same words carry no relevence in case of video games. In video games words like 'realistic' and 'hooked' are positive and 'boring' is negative.

- Due to domain specific words a classifier trained on one domain may not work on other domain.
- Domain-independent features convey similar meaning in multiple domains or in the given source and target domains.

Two methods to select domain-independent and domain-specific features :-

- **Method-1** : A naive approach is to define a frequency threshold K1 and select domain-independent features based on the frequency of occurrence in both domains (i.e. features having frequency greater than K1 in both domains). Similarly, we define a frequency threshold K2 and select domain specific features based on its frequency of occurrence in a specific domain (i.e features having frequency greater than K2 in a specific domain).

- **Method-2** : In this method we adopt the idea of Diversity and Centrality which is reflected by Domain Independent and Domain Specific features respectively. Since we traverse both source and target domain to decide Domain Independent features therefore it signifies Diversity among both domains(can also be thought as global features). Similarly, Domain Specific features only requires traversal of a particular domain, therefore it signifies centrality of features in a domain.
  For selecting domain independent features(i.e global features), we use frequency based threshold approach (as discussed in previous method).
  For selecting domain specific features(i,e local features), we use mutual information based approach.
  Mutual Information measures extent of dependence between features and domains.If a particular feature has high mutual information then it would be more dependent on a particular domain and therefore it is be more likely to be a domain specific feature.
  To select set of domain specific features in specified domain, we will calculate mutual information of each feature with respect to a particular domain and take First T% features with highest Mutual Information as Domain Specific features.

$$I(X^i;D) = \sum_{d \epsilon D} \sum_{x \epsilon X^i, x \neq 0} p(x,d) \log_2 \left(\frac{p(x,d)}{p(x)p(d)}\right)$$

	electronics	video games
positive	**Compact**; easy to operate; very *good* picture quality; looks **sharp**!	"A very *good* game! It is action packed and full of *excitement*. I am very much **hooked** on this game."
positive	I purchased this unit from Circuit City and I was very *excited* about the quality of the picture. It is really *nice* and **sharp**.	Very **realistic** shooting action and *good* plots. We played this and were **hooked**.
negative	It is also quite **blurry** in very dark settings. I will *never* buy HP again.	The game is so **boring**. I am extremely unhappy and will probably *never* buy Ubisoft again.

Table - 5.1: Table representing Domain specific and Domain Independent Features

Bipartite Feature Graph Construction

The co-occurrence relationship between domain-specific and domain-independent features is important for feature alignment across different domains. SFA uses domain-independent words as a bridge to construct a bipartite graph to model this co-occurrence relationship between domain-specific words and domain-independent words. The idea is that if two domain-specific words have connections to more common domain-independent words in the graph, they tend to be aligned together with higher probability. Similarly, if two domain-independent words have connections to more common domain-specific words in the graph, they tend to be aligned together with higher probability.

Based on the above method for selecting domain-independent and domain-specific features, we now construct a bipartite graph $G = (V_{DS} \cup V_{DI}, E)$ between them. In G, each vertex in V_{DS} corresponds to a domain-specific word in W_{DS}, and each vertex in V_{DI} corresponds to a domain-independent word in W_{DI}. An edge in E connects two vertexes in V_{DS} and V_{DI} respectively.

Each edge e_{ij} ϵ E is associated with a non-negative weight m_{ij}. The score of m_{ij} measures the relationship between word w_i ϵ W_{DS} and w_j ϵ W_{DI} in D_{src} and D_{tar}. We have used the total number of co-occurrence of w_i ϵ W_{DS} and w_j ϵ W_{DI} in D_{src} and D_{tar}. We can use other methods to estimate m_{ij}, for example, we can use the distance between w_i and w_j to adjust the score of m_{ij}.

A bipartite graph example is shown in Figure 5.1, which is constructed based on the example shown in Table 5.1.

	compact	**realistic**	**sharp**	**hooked**	**blurry**	**boring**
good	1	1	1	1	0	0
exciting	0	0	1	1	0	0
never_buy	0	0	0	0	1	1

Table - 5.2: *A co-occurrence matrix of domain-specific and domain-independent words*

Now we can use the constructed bipartite graph to model the intrinsic relationship between domain-specific and domain-independent features.

Implementation :

Some Terms :-

co-occurence - co-occurence matrix.
$wt(w_i, w_j)$ - edge weight between w_i and w_j.
$w_{specific}$ - a specific word.

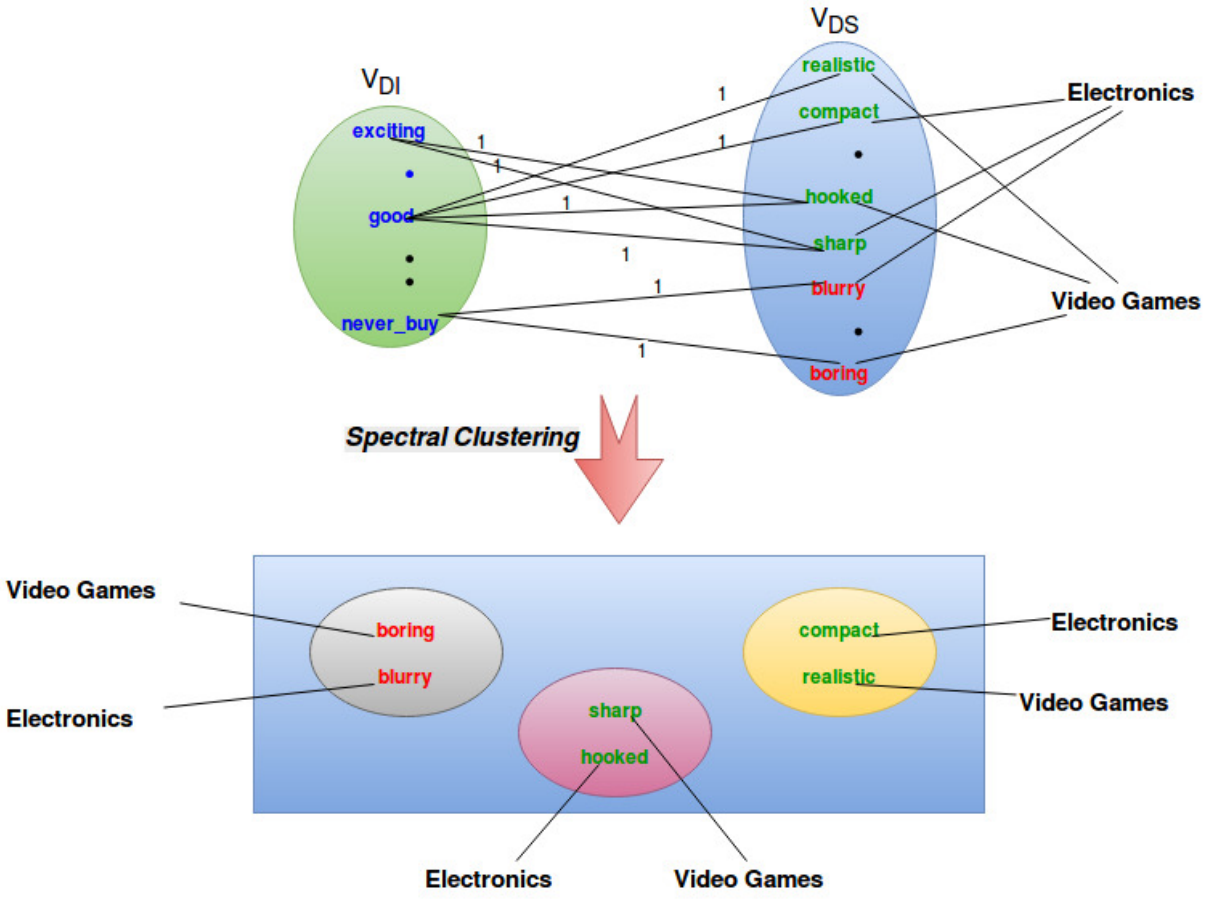

Fig - 5.1: *Bipartite graph and Spectral Clustering*[1]

$w_{independent}$ - an independent word.
$S_{article}$ - set of all articles in source corpus.
$T_{article}$ - set of all artcles in target corpus.

We represent the bipartite graph as a co-occurence matrix. In our method we have given edge weight wt to co-occurence[$w_{specific}$][$w_{independent}$] and co-occurence[$w_{independent}$][$w_{specific}$] if $w_{specific}$ and $w_{independent}$ occur together in an article.

Weighting techniques :-
We have used two weighing techniques to construct the bipartite graph of specific and independent features.

1. **Frequency based weighting** : The first method counts the total number of times $w_{specific}$ occurs with $w_{independent}$ in a article for each article of set $S_{article}$ and $T_{article}$.

$$wt(w_{specific}, w_{independent}) = \sum_{l1} \sum_{l2} (\text{count of } w_{specific} \text{ with } w_{independent} \text{ in article A})$$

l1 = A element of ($S_{article} \bigcup T_{article}$)
l2 = $w_{specific}$ and $w_{independent}$ element of article A

2. **Distance based weighting** : In this method we make use of distance between $w_{specific}$ and $w_{independent}$ in a article. The value added for each occurence of $w_{specific}$ with $w_{independent}$ and vice versa decreases with increase in distance between them.

$$wt(w_{specific}, w_{independent}) = \sum_{L1}\sum_{L2}(1000 \times (\frac{1}{(1+abs(posI(w_{independent}) - posS(w_{specific})))}))$$

L1 = A element of ($S_{article} \bigcup T_{article}$)
L2 = $w_{specific}$ and $w_{independen}$ element of article A.
posI gives position of $w_{independent}$, in article A ignoring specific words.
posS gives position of $w_{specific}$, in article A ignoring independent words.

Spectral Feature Clustering

Now, to align domain-specific features so as to reduce the gap between domains, we will need to do some type of clustering on the domain-specific and domain-independent features.

For doing this, we will now adapt a spectral clustering algorithm on the feature bipartite graph we made before.

In graph spectral theory, one of the main assumptions is:
if two nodes in a graph are connected to many common nodes, then they should be very similar (or quite related).

Spectral Clustering:

Spectral clustering technique make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions.
The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset.

It is mainly divided into two parts:

1. Feature Reduction
2. Any traditional clustering algorithm like k-means clustering.

Firstly we'll discuss the Feature Reduction part:

a). The input is Affinity/Similarity matrix (A):

$$A_{ij} = \begin{cases} w_{ij} & : weight\ of\ edge(i,j) \\ 0 & : no\ edge \end{cases}$$

b). Then we calculate Laplacian Matrix (L) of matrix A:

$$L = D^{-1/2} A D^{-1/2}$$

where $D_{ii} = \sum_j A_{ij}$

c). Calcualtion of Eigen value & Eigen vectors Sorting eigen valuesion decreasing order

λ_1 , λ_2 , λ_3 , ... , λ_n are eigen values .

μ_1 , μ_2 , μ_3 , ... , μ_n are eigen vectors .

Spectrum of Graph : Set of eigen vectors of laplacian matrix λ_2 – defines algebraic connectivity of graph i.e. more the value of λ_2, more density of connected graph.

d). Partioning of the Graph:

i). Selection of first K-eigen vectors according to sorted list of eigen values.

ii). We define $U\ \epsilon\ \mathbb{R}^{m \times k}$

iii). Normalization $U_{ij} = \frac{U_{ij}}{(\sum_j U_{ij}^2)^{1/2}}$

iv). Apply k-means clustering on U.

So, after doing all the four steps we reduced our feature set from $m \times m$ size to $m \times k$ features.

Once the feature set is reduced, then we go for clustering algorithm.

Second part : Clustering

a). It is done by applying the k-means clustering on the new reduced feature set obtained by spectral reduction.

b). This k-means algorithm clusterd the n-point into k clusters.

In our case, we assume:

1. if two domain-specific features are connected to many common domain-independent features, then they tend to be very related and will be aligned to a same cluster with high probability,
2. if two domain-independent features are connected to many common domain-specific features, then they tend to be very related and will be aligned to a same cluster with high probability.

Finally, Using the above assumptions and the weighted feature bipartite graph, we align the domain-specific features of both the domains into k clusters, where k is an input parameter.

Feature Engineering

After applying Spectral Clustring algorithm on the bipartite graph of features we have a set of clusters. Each cluster contains some domain specific and some domain independent words.As already discussed, domain specific words which convey similar sentiment will be placed in same cluster with high probability. We can use this property of clusters to modify our corpus in a manner that will make it easier for us to tranfer knowledge. Since, domain independent knowledge can be transfered without any extra effort we have to focus on domain specific words.

In our method, we will replace each domain specific word occuring in i^{th} cluster with cluster_i, (since this type of word is less likely to appear in an article) let's call this as cluster name of a domain specific word. Next step is to replace all the occurences of a word $w_{specific}$ in Source corpus and Target corpus with its corresponding cluster name. This way, we reduced the gap between domain specific words of the two domains, which is the primary objective of this algorithm. Now, each corpus contains domain independent words and cluster words (specific words replaced with their cluster names).

Using the modified Source corpus we can train our model to predict sentiments of articles in the target domain.

Proposed Models

Using above algorithm we have created three models for cross-domain sentiment analysis.
The models vary mainly in two steps :-

1. Finding domain independent and domain specific features.
2. Weighting scheme of bi-partite graph.

Model-1 :

1. Finding domain independent and domain specific features :-
   used frequency based approach to segregate independent and specific features.
2. Calculating edges of bipartite graph :-
   used frequency based weighing technique to calulate edges of bipartite graph.

Model-2 :

1. Finding domain independent and domain specific features :-
   used mutual information of features with domain as parameter to select domain independent and specific features.
2. Calculating edges of bipartite graph :-
   same as Model-1.

Model-3 :

1. Finding domain independent and specific features :-
   same as Model-2.
2. Calculating edges of bipartite graph :-
   used distance based weighting.

Dataset Description

Our model needs set of correlated datasets for better demonstration of cross domain sentiment analysis. Therefore, we picked collection of product reviews from Amazon.

The reviews are about four product domains:
Books(B), Dvds(D), Electronics(E) and Kitchen Appliances(K).

Each review is assigned a sentiment label, -1 (negative review) or $+1$ (positive review), based on the review of the products. Each domain has around 800 positive reviews and negative reviews. Using these datasets, we can create 12 pairs of source and target domains for cross-domain sentiment classification, namely:

D→B, E→B, K→B, K→E, D→E, B→E, B→D, K→D, E→D, B→K, D→K, E→K,

where the word before an arrow corresponds with the source domain and the word after an arrow corresponds with the target domain.

Experiments

Model-1 :

The different hyperparameters used in our Model-1:

a.) Domain Independent feature frequency cutoff, [IFC] The words which are present in both source and target datasets with frequency greater than this value are only considered as Independent features.
Range:- [10, 15, 20, 25, 30].

b.) Domain Specific feature frequency cutoff, [SFC] The words which are present in one or both of the datasets with a frequnency less than this value are only considered as specific features.
Range:- [10, 15, 20, 25, 30].

c.) No. of clusters assumed for clustering. [NCC]
Range:- [10, 20, 30, 40, 50].

So for each of the above combination of datasets we tried all $(5 \times 5 \times 5)$ 125 combinations.

Model-2 :

The way we did variations for different hyperparameters in our Model-2 is as follows:

a.) Domain Independent feature frequency cutoff, [IFC] The words which are present in both source and target datasets with frequency greater than this value are only considered as Independent features.
Range:- [10, 15, 20, 25, 30].

b.) Mutual information of features with a domain is calculated. The result is stored in two seperate lists , source features list and target features list. The lists are sorted in descending order of mutual information. Top features from each list are selected as specific features belonging to corresponding domain.
Range:- top x percent features are selected x element [33,12.5,7.7,5.6,4.35]

c.) No. of clusters assumed for clustering. [NCC]
Range:- [10, 20, 30, 40, 50].

So for each of the above combination of datasets we tried all $(5 \times 5 \times 5)$ 125 combinations.

Model-3 :

In method 3 all hyperparameters are same as that of Model-2. In Model-3 the we have used a distance based weighting approach to calculate edge weights of bipartite graph.

Source	Target	No. of Clusters	Specific Cutoff	Independent Cutoff	Accuracy
electronics	kitchen & housewares	20	10	10	80.75
kitchen & housewares	electronics	10	30	20	79.4
books	dvd	30	10	25	78.7
dvd	books	30	15	25	77.85
dvd	kitchen & housewares	20	10	10	71.8
kitchen & housewares	dvd	50	20	10	71.5
electronics	dvd	40	30	25	70.7
dvd	electronics	40	10	10	68.7
electronics	books	50	30	25	68.1
books	kitchen & housewares	30	25	15	67.75
kitchen & housewares	books	40	25	15	67.15
books	electronics	30	10	25	66.8

Table - 8.1: Table for Model-1 consisting of Ideal hyperparameters after parameter tuning and corresponding Accuracy for each combinations of different datasets

Source	Target	No. of Clusters	Specific Cutoff	Independent Cutoff	Accuracy
electronics	kitchen & housewares	20	3	15	80.65
kitchen & housewares	electronics	15	3	10	79.5
books	dvd	15	3	20	78.7
dvd	books	35	8	30	78.2
dvd	kitchen & housewares	30	13	10	72
kitchen & housewares	dvd	35	23	10	71.4

electronics	dvd	25	3	15	71.3
dvd	electronics	30	3	15	69.1
electronics	books	30	8	10	68.3
books	electronics	30	3	20	67.9
books	kitchen & housewares	25	13	10	67.7
kitchen & housewares	books	15	3	10	67.4

Table - 8.2: Table for Model-2 consisting of Ideal hyperparameters after parameter tuning and corresponding Accuracy for each combinations of different datasets

Source	Target	No. of Clusters	Specific Cutoff	Independent Cutoff	Accuracy
electronics	kitchen & housewares	30	8	10	80.55
kitchen & housewares	electronics	35	3	20	79.55
books	dvd	20	23	25	78.9
dvd	books	35	3	30	77.95
dvd	kitchen & housewares	25	3	10	72.45
kitchen & housewares	dvd	35	23	10	71.25
electronics	dvd	10	3	25	70.85
dvd	electronics	30	8	30	69.15
books	kitchen & housewares	20	8	10	69.1
electronics	books	30	8	25	68.75
kitchen & housewares	books	30	23	15	68.05
books	electronics	30	3	20	67.6

Table - 8.3: Table for Model-3 consisting of Ideal hyperparameters after parameter tuning and corresponding Accuracy for each combinations of different datasets

Source	Target	Method 1	Method 2	Method 3
electronics	kitchen & housewares	80.75	80.65	80.55

kitchen & housewares	electronics	79.4	79.5	79.55
books	dvd	78.7	78.7	78.9
dvd	books	77.85	78.2	77.95
dvd	kitchen & housewares	71.8	72	72.45
kitchen & housewares	dvd	71.5	71.4	71.25
electronics	dvd	70.7	71.3	70.85
dvd	electronics	68.7	69.1	69.15
books	kitchen & housewares	68.1	68.3	69.1
electronics	books	67.75	67.9	68.75
kitchen & housewares	books	67.15	67.7	68.05
books	electronics	66.8	67.4	67.6

Table - 8.4: Comparison of Accuracy obtained by using different models proposed in the report on different combinations of datasets

Graphs

For each method we implemented, there are three graphs which shows how the accuracy (best) changes with respect to variation of the 3 hyperparameters, namely

1 Accuracy vs. No. of clusters.

2 Accuracy vs. Specific threshold value.

3 Accuracy vs. Independent threshold value.

Each graph shows the accuracy for all the datasets, (datasets represented by different colors).
Finally we plotted a graph which shows how accuracy for a specific pair of dataset changes with respect to different methods.

Model-1 :

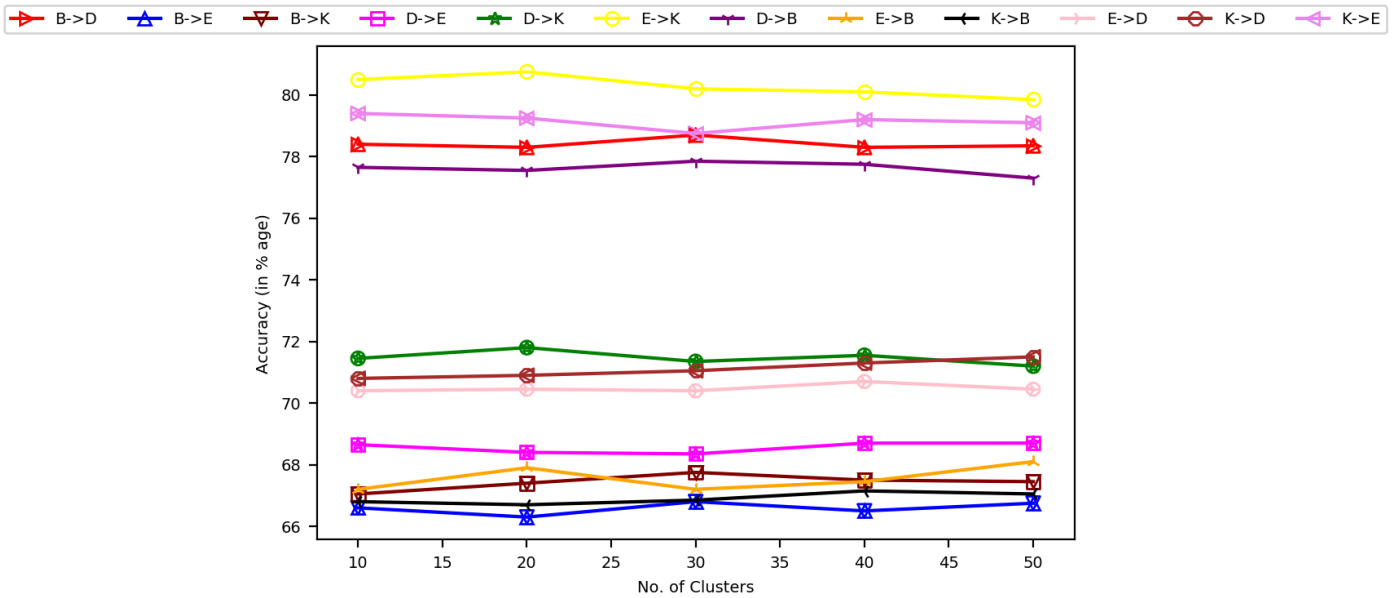

Fig - 9.1: *Accuracy(best) vs Number of Clusters*

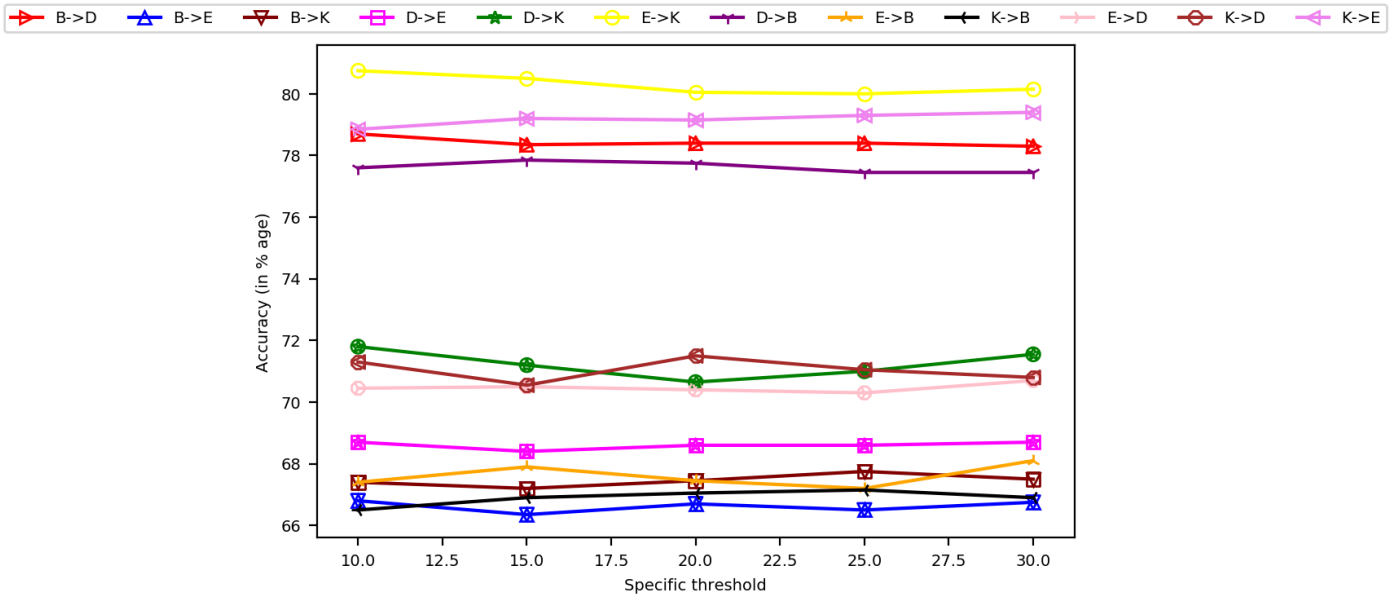

Fig - 9.2: *Accuracy(best) vs Specific Threshold*

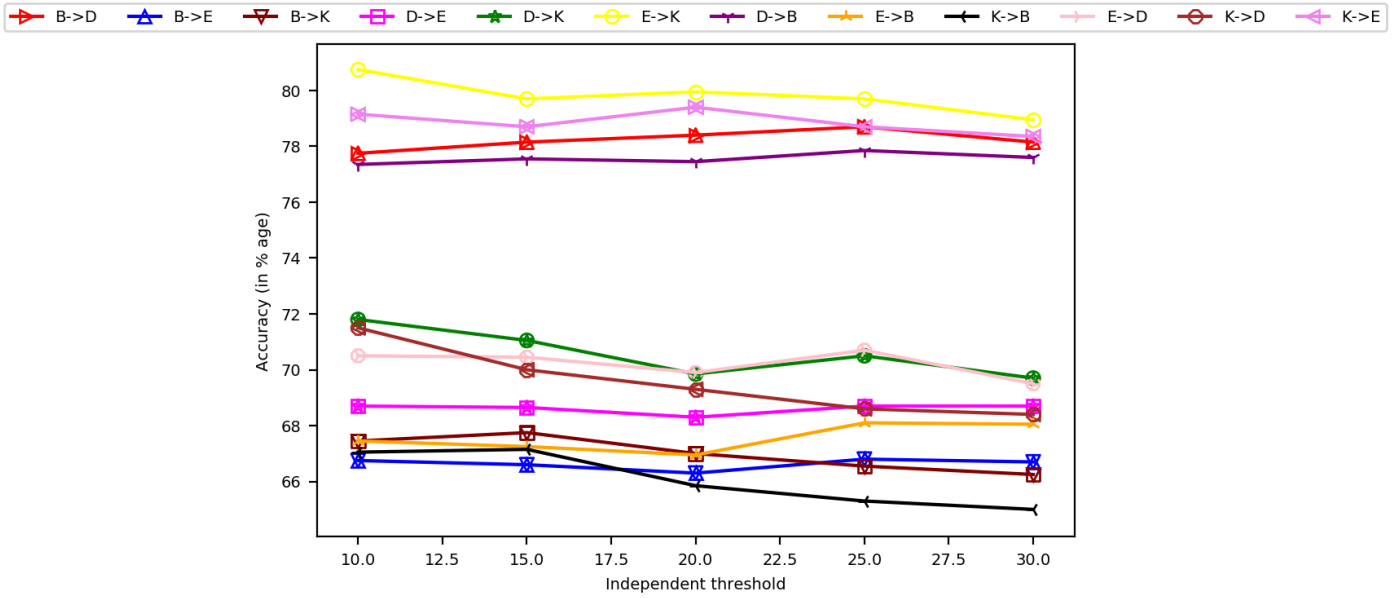

Fig - 9.3: *Accuracy(best) vs Independent Threshold*

Model-2 :

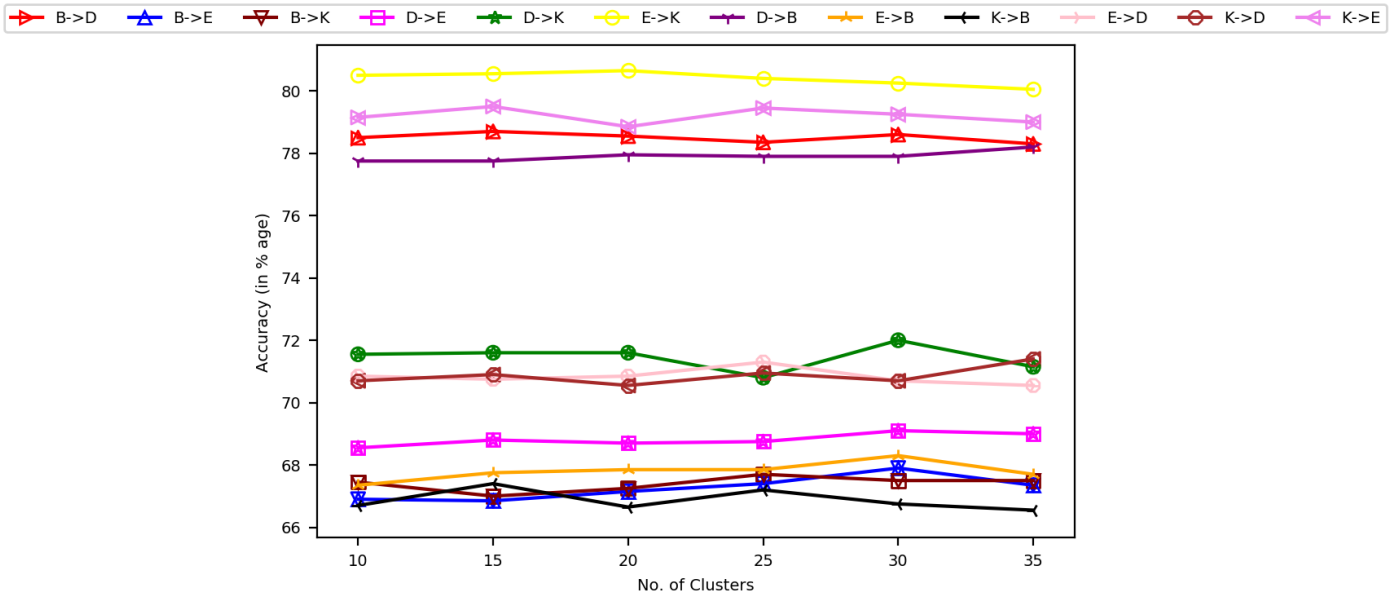

Fig - 9.4: *Accuracy(best) vs Number of Clusters*

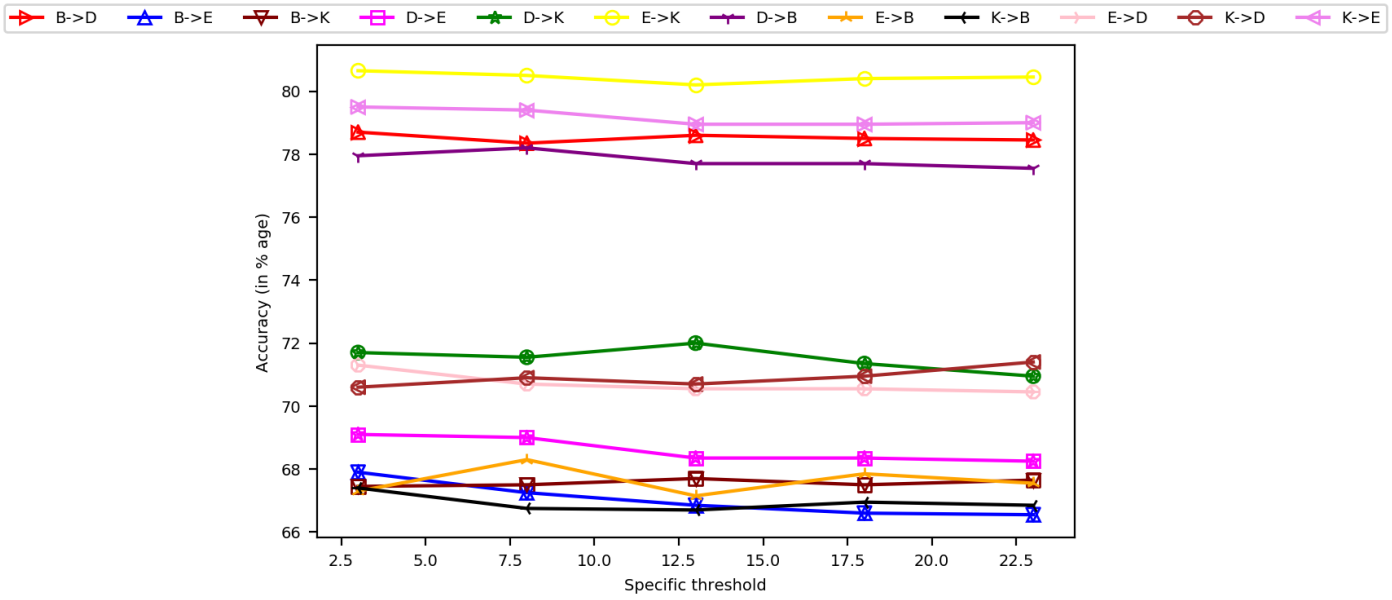

Fig - 9.5: *Accuracy(best) vs Specific Threshold*

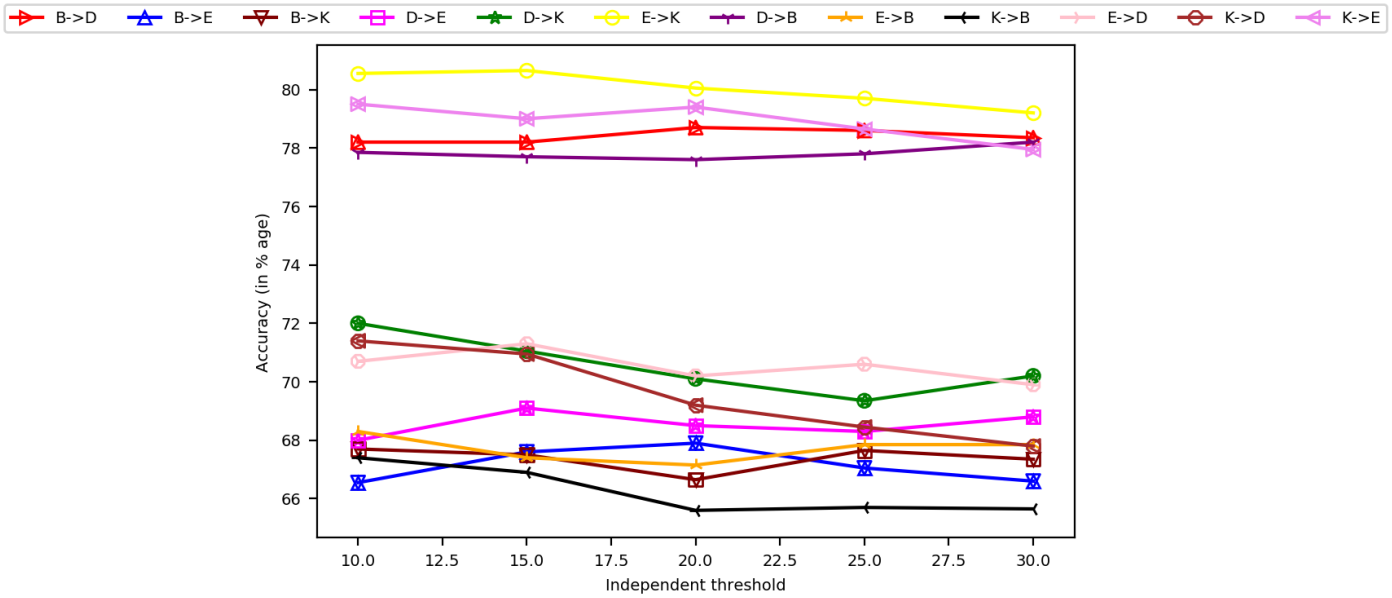

Fig - 9.6: *Accuracy(best) vs Independent Threshold*

Model-3 :

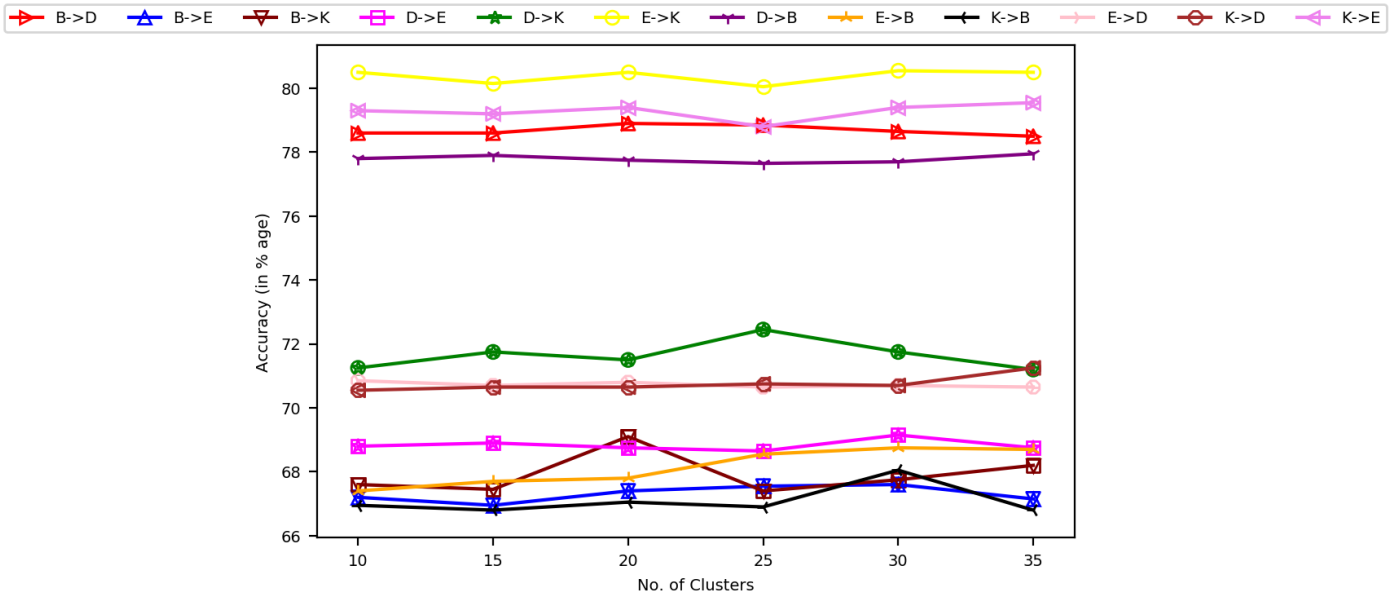

Fig - 9.7: *Accuracy(best) vs Number of Clusters*

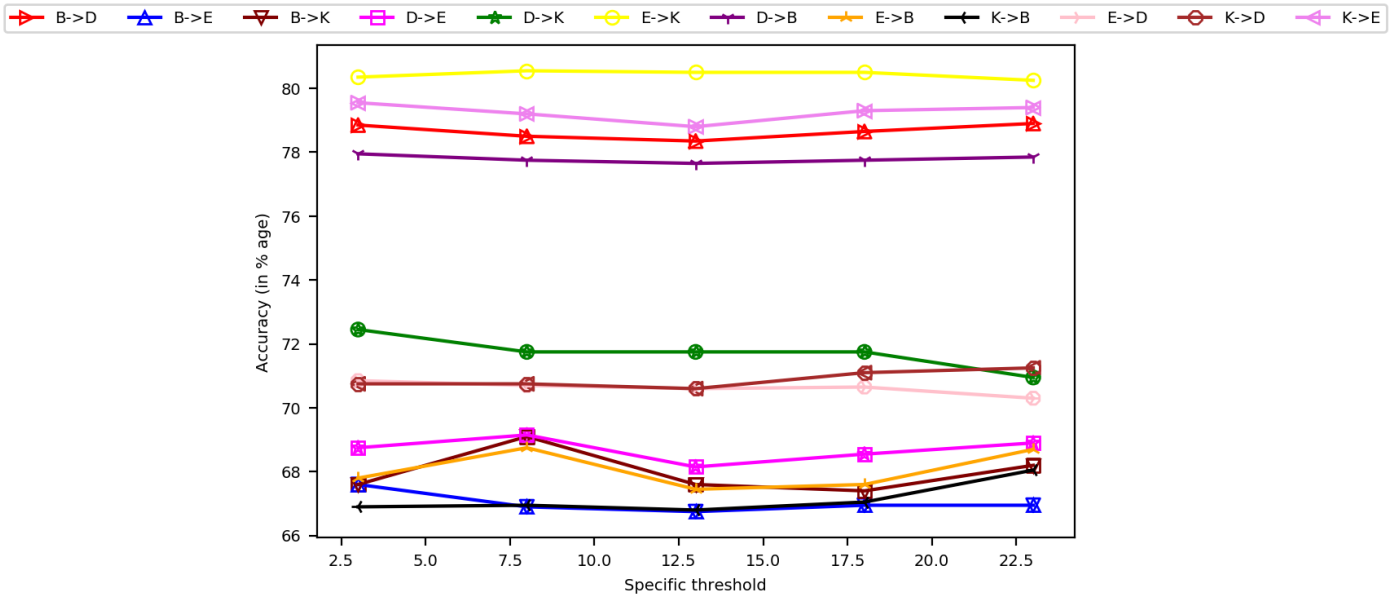

Fig - 9.8: *Accuracy(best) vs Specific Threshold*

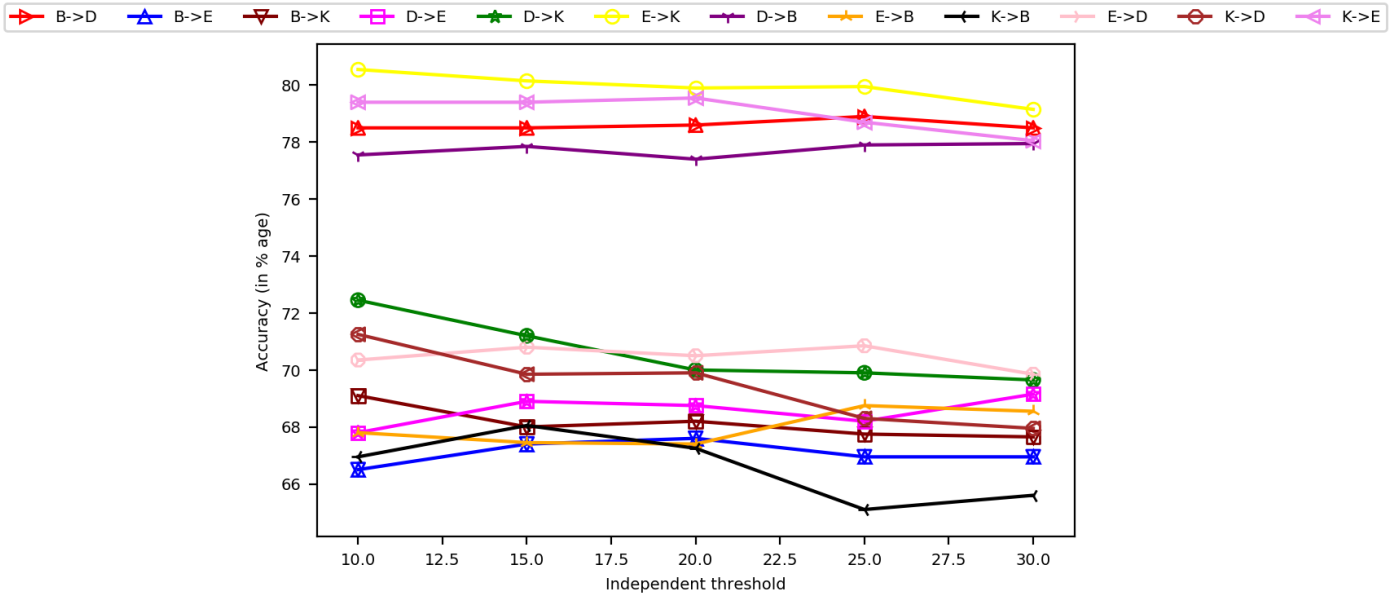

Fig - 9.9: *Accuracy(best) vs Independent Threshold*

Methods Comparison

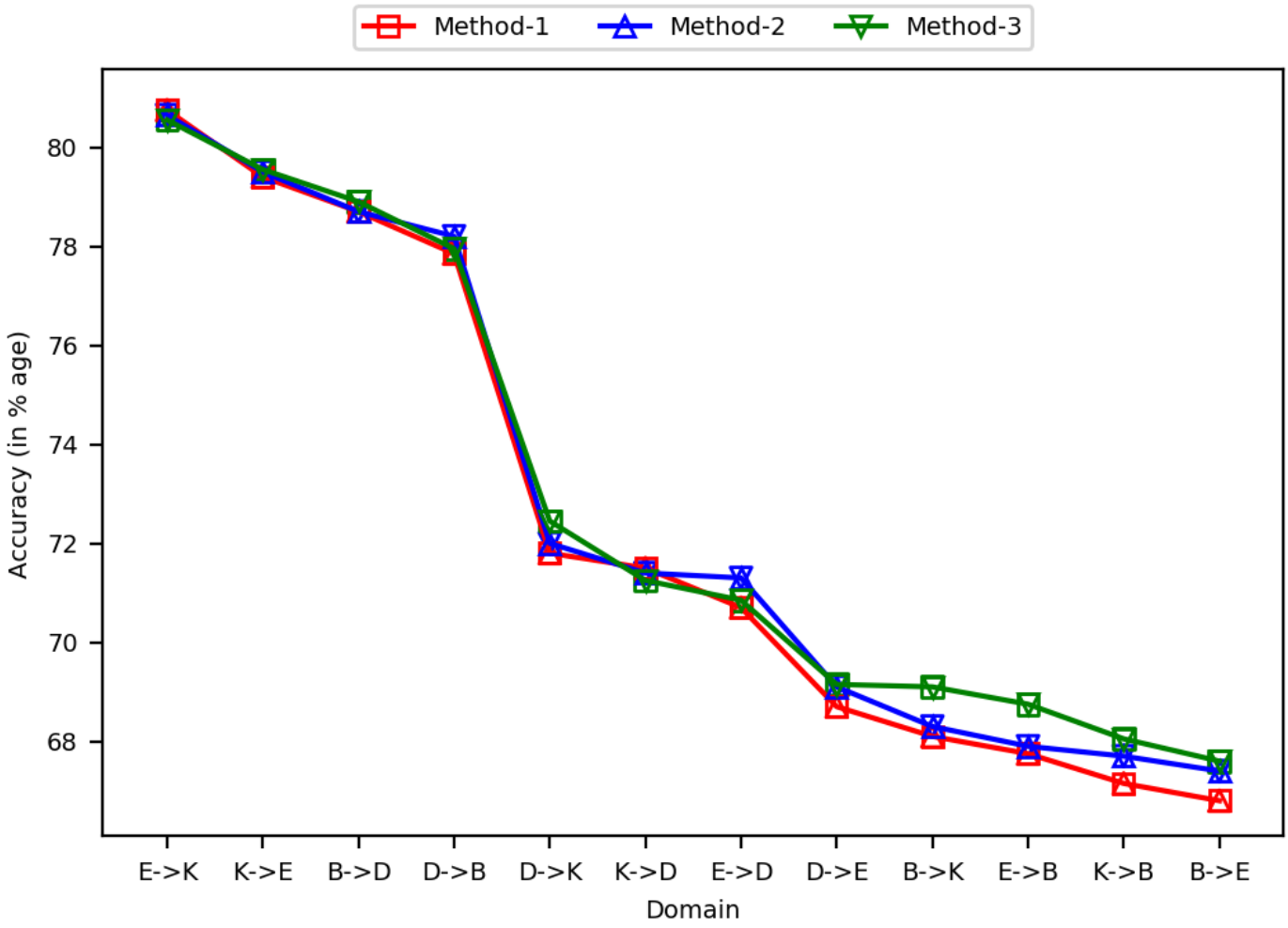

Fig - 9.10: *Accuracy(best) of each domain pair for all methods*

Observations :

I. After testing our methods with different combinations of domains(i.e Books → Dvds, Dvds → electronics.etc) we observed that :

   ◇ For Model 1 :

      1. We are obtaining best accuracy for various combinations of domains when the number of clusters lie in the range 25-30(Fig-9.1).
      2. When specific threshold lies between 20-25, we get the best accuracy for given domains(Fig-9.2).
      3. When independent threshold lies between 10-15 & 23-28, the best accuracy is obtained for given combinations of domains(Fig-9.3).

   ◇ For Model 2 :

      1. As evident from Fig-9.4 , we get best accuracy when mutual information based specific percentage cuttoff is the range 12.5% 33 %.
      2. The optimum independent threshold lies between 10-15 and 20-25 (Fig-9.6).
      3. The optimum cluster length lies between 20-30 (Fig-9.4).

   ◇ For Model 3 :

      1. As evident from Fig-9.8 , we get best accuracy when mutual information based specific percentage cuttoff is the range 4%-12.5%.
      2. The optimum independent threshold lies between 10-20 (Fig-9.9).
      3. The optimum cluster length lies between 20-30 (Fig-9.7).

II. From table(Table-8.4), it can be observed that method 3 has best accuracy for 8 out of 12 combinations of domains though the difference between the accuracies is quite less.

III. As observed in Fig-9.1 to 9.9, there is a significant gap between accuracies of some combination of domains.

Conclusions

(i) More related are the domains, more is the accuracy. for instance Books and Dvds (78%) are more related to each other than books and eletronics therefore they have better accuracy than books and electronics (67%). This explains the gap between accurancies for some combination of domains in fig.

(ii) The best method for cross domain sentiment analysis on given dataset is Method 3 as it giving best results for 8 out of 12 combinations of domains.

(iii) The optimum cluster length always remains in range 20-30 irrespective of which method is used.

(iv) The optimum mutual information based specific percentage cuttoff drops from 12.5% - 33% to 4.5% - 12.5% when we shift from method 2 to method 3 respectively.

Softwares Used :

Spyder 3 (Scientific PYthon Development EnviRonment)

Language Used :

Python 3.5

Libraries Used :

- pandas (Python Data Analysis Library) for inputting the Dataset.
- NumPy for creating the co-occurence matrix (i.e. a large multi-dimensional matrix).
- The module pyplot of matplotlib library is used for graph plotting re library for the usage of Regular Expressions for text cleaning.
- Stopwords list from the 'corpus' package of nltk (Natural Language Processing Toolkit) data package.
- PorterStemmer algorithm from the package nltk.stem.porter for stemming of words.
- CountVectorizer class from sklearn.feature_extraction.text submodule for building the matrix (sparse) of word counts from text documents.
- TfidfTransformer class from sklearn.feature_extraction.text submodule to convert the count matrix (sparse) to a matrix of TF_IDF features.
- SpectralClustering class from sklearn.cluster module for the Spectral Clustering of the bipartite graph (represented as the co-occurence matrix).
- train_test_split function from sklearn.cross_validation module for splitting the whole Dataset into random train and test sunsets.
- SVC class from sklearn.svm module for classification of the test set.

References

[1] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 751–760. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772767

[2] C.-K. Lin, Y.-Y. Lee, C.-H. Yu, and H.-H. Chen, "Exploring ensemble of models in taxonomy-based cross-domain sentiment classification," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '14. New York, NY, USA: ACM, 2014, pp. 1279–1288. [Online]. Available: http://doi.acm.org/10.1145/2661829.2662071

[3] F. Wu, Y. Huang, and Z. Yuan, "Domain-specific sentiment classification via fusing sentiment knowledge from multiple sources," *Information Fusion*, vol. 35, pp. 26 – 37, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253516300653

[4] A. R. Nelakurthi, H. Tong, R. Maciejewski, N. Bliss, and J. He, *User-guided Cross-domain Sentiment Classification*, pp. 471–479. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611974973.53

[5] F. Wu, S. Wu, Y. Huang, S. Huang, and Y. Qin, "Sentiment domain adaptation with multi-level contextual sentiment knowledge," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16. New York, NY, USA: ACM, 2016, pp. 949–958. [Online]. Available: http://doi.acm.org/10.1145/2983323.2983851