Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach

ADM Assignment#2

Submitted to: Poonam Goyal

Submitted by: Aastha Sanghi (2016h112159P)

M. sushmitha malae (2016h112173P)

**2017**

Table of Contents

[Introduction 2](#_Toc481213244)

[Domain Adaption 2](#_Toc481213245)

[Dataset 2](#_Toc481213246)

[Protocol Used in Base Paper 2](#_Toc481213247)

[Neural Network 2](#_Toc481213248)

[SDA 3](#_Toc481213249)

[Components 3](#_Toc481213250)

[Pre-processing 4](#_Toc481213251)

[Feature Extraction 4](#_Toc481213252)

[Classification 4](#_Toc481213253)

[Protocol Proposed 4](#_Toc481213254)

[Neural Network 4](#_Toc481213255)

[CNN 4](#_Toc481213256)

[SDA 5](#_Toc481213257)

[Components 5](#_Toc481213258)

[1. Feature Selection 6](#_Toc481213259)

[2. Feature Extraction and Classification 6](#_Toc481213260)

[Training 7](#_Toc481213261)

[Testing 8](#_Toc481213262)

[Results 8](#_Toc481213263)

# Introduction

With the rise of social media such as blogs and social networks, reviews, ratings and recommendations are rapidly proliferating; being able to automatically filter them is a current key challenge for businesses looking to sell their wares and identify new market opportunities. Sentiment classification has its own significance when applied to reviews available. Reviews can span over different domains. Thus, major focus is on cross-domain classification i.e. a system trained on labeled reviews from one source domain but deployed on another.

# Domain Adaption

Domain adaptation considers the setting in which the training and testing data are sampled from different distributions. When more than one source distribution is available, the problem is referred to as multi-source domain adaptation. There are several contexts of domain adaptation. They differ in the information considered for the target task.

1. The *unsupervised domain adaptation*: the learning sample contains a set of labeled source examples, a set of unlabeled source examples and an unlabeled set of target examples.
2. The *semi-supervised domain adaptation*: in this situation, we also consider a "small" set of labeled target examples.
3. The *supervised domain adaptation*: all the examples considered are supposed to be labeled.

# Dataset

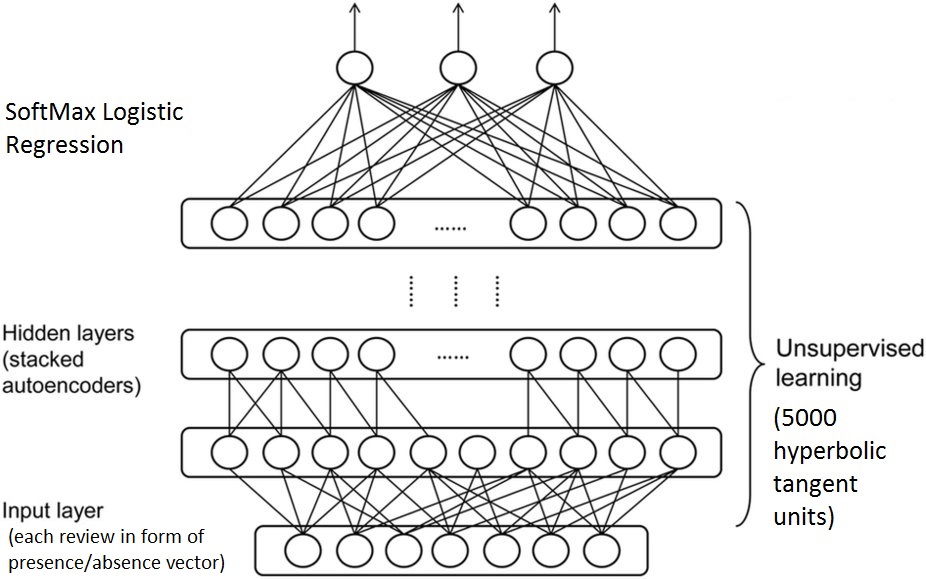
Amazon data: The data set proposes more than 340,000 reviews regarding 22 different product types and for which reviews are labeled as either positive or negative. Since this data set is heterogeneous, heavily unbalanced and large-scale, a reduced set has been released. The reduced data set contains 4 different domains: Books, DVDs, Electronics and Kitchen appliances. There are 1000 positive and 1000 negative instances for each domain, as well as a few thousand unlabeled examples. The positive and negative examples are also exactly balanced.

# Protocol Used in Base Paper

## Neural Network

### SDA

Stacked Denoising Auto-encoder. Below figure gives a abstract picture of SDA applied on Amazon reviews.



Input to SDA

**Auto-encoder**: An auto-encoder is comprised of an encoder function h(·) and a decoder function g(·), typically with the dimension of h(·) smaller than that of its argument. The reconstruction of input x is given by r(x) = g(h(x)), and auto-encoders are typically trained to minimize a form of reconstruction error loss(x, r(x)). Examples of reconstruction error include the squared error or the Kullback- Liebler divergence between elements of x and elements of r(x).

**Stacked auto-encoders:** Once an auto-encoder has been trained, one can stack another auto-encoder on top of it, by training a second one, which sees the encoded output of the first one as its training data.

**Denoising stacked auto-encoder**: The input vector x is stochastically corrupted into a vector x’, and the model is trained to denoise, i.e., to minimize a denoising reconstruction error loss(x, r(x’)).

## Components

### Pre-processing

Each review text is treated as a bag-of-words and transformed into binary vectors encoding the presence/absence of unigrams and bigrams. For computational reasons, only the 5000 most frequent terms of the vocabulary of unigrams and bigrams are kept in the feature set.

### Feature Extraction

A higher-level feature extraction is learnt in an unsupervised fashion from the text reviews of all the available domains using a Stacked Denoising Auto-encoder (SDA) with rectifier units (i.e. max(0, x)) for the code layer. The SDA is learnt in a greedy layer-wise fashion using stochastic gradient descent and the training criterion is the Kullback-Liebler divergence.

### Classification

SVM a linear classifier is trained on the transformed labeled data of the source domain. SVM, with squared hinge loss is tested on the target domain(s). SVM takes data for which features have been transformed by the preprocessing component.

# Protocol Proposed

## Neural Network

### CNN

Convolutional neural networks consist multiple layer mainly for text classification that are:

1. **EMBEDDING LAYER**

The first layer we define is the embedding layer, which maps vocabulary word indices into low-dimensional vector representations.

1. **CONVOLUTION**

A convolution operation involves a filter, which is applied to a window of some h words to produce a new feature.

1. **MAX-POOLING**

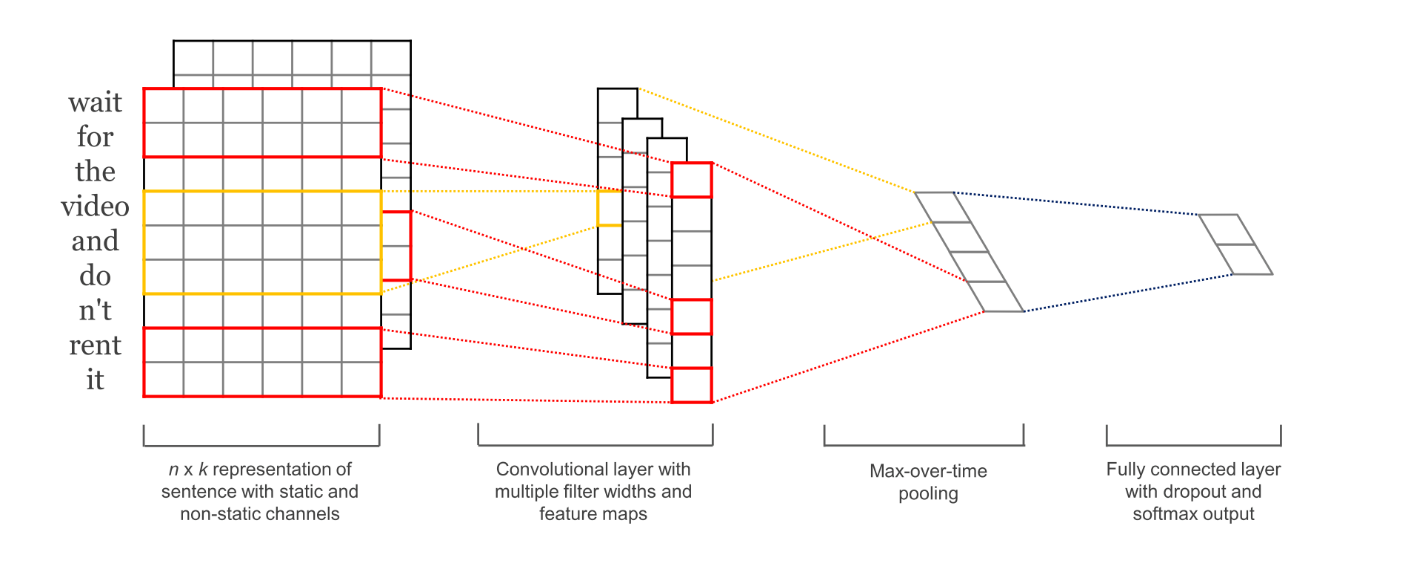
Performing max-pooling over the output of a specific filter size leaves us with a vector. This is essentially a feature vector, where the last dimension corresponds to our features. Once we have all the pooled output tensors from each filter size we combine them into one long feature vector.

1. **Dropout layer**

A dropout layer stochastically “disables” a fraction of its neurons. This prevent neurons from co-adapting and forces them to learn individually useful features.

1. **Softmax ouput**

Using the feature vector from max-pooling (with dropout applied) we can generate predictions by doing a matrix multiplication and picking the class with the highest score. We apply a softmax function to convert raw scores into normalized probabilities.

****

### SDA

Stacked denoising auto encoder as mentioned above is comprised of an encoder function and a decoder.The reconstruction of input is done and auto-encoders are typically trained to minimize a form of reconstruction error loss(x, r(x)).

## Components

### Feature Selection

Feature selection techniques can be organized into three categories:

* filter
* wrapper
* embedded

Among the three categories, the filter-based technique is the most suitable one since it is simple, fast, and independent from classifiers. Examples of the filter-based technique are Information Gain, χ2 (Chi-squared), Mutual Information, t-test, F-measure etc. Out of these the Chi-squared filter-based technique as the most effective way to perform the feature reduction in text analysis problems.

**χ2 (Chi-squared) Feature Selection:** The higher score on χ2, the more dependent between the feature term and class, and therefore considered important. Consequently, the top rank features with most χ2 scores are chosen as a set of features for the text classification.

Feature selection is performed on the whole vocabulary for both the domain i.e. source as well as target domain.

### Feature Extraction and Classification

**Trained layers from input set domain S are frozen and passed to domain set** T

Six layers are considered that are:

1. **EMBEDDING LAYER**

The first layer we define is the embedding layer, which maps vocabulary word indices into low-dimensional vector representations.

1. **CONVOLUTION**

A convolution operation involves a filter, which is applied to a window of words to produce a new feature. Let g is the result of applying the nonlinearity to the convolution output.

1. **MAX-POOLING**

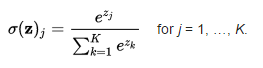
Performing max-pooling over the output of a specific filter size leaves us with a vector. This is essentially a feature vector, where the last dimension corresponds to our features. Once we have all the pooled output tensors from each filter size we combine them into one long feature vector.

1. **SDA(1,2)**

These layers are fine-tuning layer to learn the features better which are extracted from first three layers. Each layer learns a non-linear mapping where is RELU(rectified unit)

1. **SDA3**

This is the ouput layer which predicts the label of the given review. It uses softmax function for the ouput.



### Training

T1: After feature selection, above six layers are trained on input data of source domain and last three layers are fine-tuned for the features in domain S

T2: First Three layers (E,C,M-P) which are trained on input data of source are frozen and input data applied as target(T) domain and last three layers are fine-tuned for the features in domain T

### Testing

Label prediction happens on T2 model for the target domain.

# Results

**Feature Selection**

#Reviews vs. Vocab size

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #Domain Source | #Domain Target | #Reviews | Vocabulary size | Vocab after FS |
| 300 | 300 | 600 | 12627 | 8430 |
| 600 | 600 | 1200 | 18182 | 11495 |
| 1000 | 1000 | 2000 | 23435 | 14290 |
| 1500 | 1500 | 3000 | 28401 | 16789 |
| 2000 | 2000 | 4000 | 32629 | 18987 |

Threshold vs Vocab Size

|  |  |  |
| --- | --- | --- |
| #Reviews | Threshold | Vocab after FS |
| 4000 | 0.2 | 18987 |
| 4000 | 0.3 | 18434 |
| 4000 | 0.31 | 18421 |
| 4000 | 0.315 | 18420 |
| 4000 | 0.32 | 6674 |
| 4000 | 0.325 | 6669 |
| 4000 | 0.35 | 6600 |
| 4000 | 0.4 | 6404 |

Benchmark Dataset:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Domain | Train Size | | Test Size | |
|  | Positive | Negative | Positive | Negative |
| Kitchen | 200 | 200 | 100 | 100 |
| Electronics | 200 | 200 | 100 | 100 |
| DVDs | 200 | 200 | 100 | 100 |
| Books | 200 | 200 | 100 | 100 |

Network parameter change

|  |  |
| --- | --- |
| Epochs | cross Entropy loss |
| 2 | 0.356 |
| 3 | 0.327 |
| 4 | 0.318 |
| 5 | 0.292 |
| 6 | 0.29 |

|  |  |
| --- | --- |
| BatchSize | Cross Entropy loss |
| Equal to dataset | 0.18 |
| half the dataset | 0.241 |
| quarter the dataset | 0.263 |

Cross-Domain predictions

**Accuracy**

Kitchen-> DVDs

|  |  |  |
| --- | --- | --- |
| Source (%) | Target(%) | Accuracy(%) |
| 75 | 25 | 72 |
| 50 | 50 | 83 |
| 25 | 75 | 87 |

Kitchen->Books

|  |  |  |
| --- | --- | --- |
| Source (%) | Target(%) | Accuracy(%) |
| 75 | 25 | 67 |
| 50 | 50 | 86 |
| 25 | 75 | 92 |

**Transfer Loss**

|  |  |
| --- | --- |
| Source->Domain | Loss Transfer |
| K->D | 1- |
| K->E | -1 |
| K->B | 5 |