Domain Adaptation for Disaster-related Twitter Data

by

Oleksandra Sopova

B.S., National University of Kyiv-Mogyla Academy, 2015

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

Machine learning is the subfield of Artificial intelligence that gives computers the ability to learn without being explicitly programmed, as it was defined by Arthur Samuel - the American pioneer in the field of computer gaming and artificial intelligence who was born in Emporia, Kansas.

Supervised Machine Learning is focused on building predictive models given labeled training data. Data may come from a variety of sources, for instance, social media networks.

In our research, we use Twitter data, specifically, user-generated tweets about disasters such as floods, hurricanes, terrorist attacks, etc., to build classifiers that could help disaster management teams identify useful information.

A supervised classifier trained on data (*training data*) from a particular domain (i.e. disaster) is expected to give accurate predictions on unseen data (*testing data*) from the same domain, assuming that the training and test data have similar characteristics. Labeled data is not easily available for a current *target* disaster.

However, labeled data from a prior *source* disaster is presumably available, and can be used to learn a supervised classifier for the target disaster.

Unfortunately, the source disaster data and the target disaster data may not share the same characteristics, and the classifier learned from the source may not perform well on the target. Domain adaptation techniques, which use unlabeled target data in addition to labeled source data, can be used to address this problem.

We study single-source and multi-source domain adaptation techniques, using Nave Bayes and extreme gradient boosting classifiers.

Experimental results on Twitter datasets corresponding to six disasters show that domain adaptation techniques improve the overall performance as compared to basic supervised learning classifiers.

Domain adaptation is crucial for many machine learning applications, as it enables the use of unlabeled data in domains where labeled data is not available.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Dr. Doina Caragea for her excellent advice, guidance, valueable comments, suggestions and inspiration throughout my Masters program, and for providing the freedom in doing research and exploring various ideas.

I would like to thank Dr. Amtoft Torben for being a member of my M.S. committee, and for his excellent courses on algorithms and software specifications, which I enjoyed greatly and which were very beneficial to me.

I would like to thank Dr. Mitchell L. Neilsen, for being a member of my M.S. committee, and for providing great advice and support throughout his classes, and for creating a very welcoming learning environment.

I am also grateful to my family, friends and colleagues for their support and encouragement.

Dedication

I dedicate this work to my family.

Chapter 1

Introduction

In this chapter, we first introduce the basic terminology in the field of Machine Learning that we use thoughout this document in Section 1.1, and then provide background on disaster management in Section 1.2.

We state the main problem addressed in Section 1.3, where we also give a high-level overview of the approaches used in this work.

1.1 Basic Terminology

An agent is learning if it improves its performance on future tasks after making observations about the world, as it is defined in [Russell and Norvig, 2009]. The examples of tasks may include:

- identify a given email as spam or non-spam
- predict housing prices for a given location
- determine if there is a specific object in a given image
- categorize news articles into topics such as politics, sports, entertainment, etc

Machine Learning algorithms use feature based representations for instances, where each instance is represented using a collection of features $f_1, f_2, \cdots, f_n$ [Mitchell, 1997]. An in-

stance is a single object of the world from which a model will be learned, or on which a model will be used (e.g., for prediction). In most machine learning work, instances are described by feature vectors; some work uses more complex representations (e.g., containing relations between instances or between parts of instances) [Kohavi and Provost, 1998]. For example, an instance of the task "identify a given email as spam or non-spam" may be a text of an email represented as bag-of-words [Mitchell, 1997]. In this work, we use the words "instance" and "example" interchangeably.

Two major types of learning are distinguished: supervised and unsupervised learning.

In supervised learning the agent is given a training set of N examples, which could be seen as input-output pairs $(x_1, y_1), (x_2, y_2), \cdots (x_N, y_N)$, where each y_j was generated by an unknown function $y = f(x)$. The task is to should discover a function h that approximates the true function f [Russell and Norvig, 2009]. Identifying a given email as spam or non-spam is an example of a supervised learning task since training a model requires labeled instances, i.e. emails marked as spam and non-spam.

In unsupervised learning the agent learns patterns in the input even though no explicit feedback is supplied [Russell and Norvig, 2009]; essentially, it means that only $(x_1), (x_2), \cdots (x_N)$ are provided. Categorizing news articles into topics such as politics, sports, entertainment, etc is an example of an unsupervised learning task since it requires finding similarity between different news articles and clustering them together, with no prior labels provided.

A classifier, or a classification model, is defined as a mapping from unlabeled instances to (discrete) classes. Classifiers have a form (e.g., decision tree) plus an interpretation procedure (including how to handle unknowns, etc.). Some classifiers also provide probability estimates (scores), which can be thresholded to yield a discrete class decision thereby taking into account a utility function [Kohavi and Provost, 1998].

In this work, we primarily focus on the first type of learning in the attempt to take advantage of labeled data. We also explore unsupervised methods to make use of unlabeled data.

1.2 Background on Disaster Management

Social media have become an integral part of disaster response. Twitter is one of the social media networks that can fill the void in areas where cell phone service might be lost, and where people look to resources to keep informed, locate loved ones, notify authorities and express support [Maron, 2013].

For instance, the Federal Emergency Management Agency (FEMA) wrote in its 2013 National Preparedness report that during and immediately following Hurricane Sandy, users sent more than 20 million Sandy-related Twitter posts, or tweets, despite the loss of cell phone service during the peak of the storm. New Jerseys largest utility company, PSE&G, said at the subcommittee hearing that during Sandy they staffed up their Twitter feeds and used them to send word about the daily locations of their giant tents and generators [Maron, 2013].

Following the Boston Marathon bombings, one quarter of Americans reportedly looked to Facebook, Twitter and other social networking sites for information, according to The Pew Research Center. The sites also formed a key part of the information cycle: when the Boston Police Department posted its final "CAPTURED!!!" tweet of the manhunt, more than 140,000 people retweeted it [Maron, 2013].

Furthermore, the National Disaster Management Authority (NDMA) spearheads an integrated approach to disaster management for the Government of India. They use Twitter to receive reports of damage from the field, and crowdsource the data to get sophisticated insights into what is happening in a region. Often the scale of events is so big, plotting data from Twitter helps to prioritize areas most affected. Twitter also helps the government communicate to the people affected what relief is available to them and where they can go to receive it [Kaul, 2016].

1.3 Problem Definition

According to the members of the National Disaster Management Authority (NDMA), some of the challenges of using social media during disasters include tweaking the strategy in real time as the disaster you planned for is not the disaster that happens, using the right hashtags as part of a good Twitter strategy', and devising ways to deal with misinformation and rumours [Kaul, 2016].

Machine learning methods may serve as an efficient solution to identify relevant tweets about disasters [Ashktorab et al., 2014]. In fact, supervised machine learning methods have been used extensively because of the availability of labeled training data — tweets about previous disasters — that have been labeled by crowdsourcing, and thus, can be used to train a classification model [Starbird et al., 2010]. Furthermore, since tweets are primarily texts, natural language processing (NLP) methods for disaster management have been also researched by [Sakaki et al., 2010] and [Terpstra, 2012].

However, there are still many challenges in using relevant data from Twitter to help disaster response teams save peoples lives and property [Mendoza et al., 2010]. One of the major challenges is that for a current on-going disaster, no labeled data is available. Obviously, labeling data is an expensive, time-consuming and error-prone process. Thus, using supervised learning methods to aid disaster response may not be time-efficient. Yet, labeled data for a previous disaster may be available. We call a previous disaster data *source* and current on-going disaster data *target*. In addition, even though labeled target data is not available, unlabeled target data may still be available, and we explore its usage throughout this work.

We define the task as follows: given labeled source tweets, train a model to classify target tweets as *relevant/not relevant* i.e. *about disaster/not about disaster*. Yet, given that the distributions of source and target data are generally different, our model may not perform well.

One of the key challenges is adapting the classifier to perform well on a target unlabeled domain. The process of adapting a classifier to make predictions on an unseen domain

where labeled data is unavailable is called Domain Adaptation. In supervised machine learning, the general assumption is that both training and testing data come from the same distribution. This holds true for data coming from the saame domain. However, in real-world classification problems training and testing data may come from different domains. As a result, the performance of a classifier may drop significantly. Thus, it becomes essential to adapt the classifier trained on one domain to give accurate prediction on another domain.

We raise the following questions, which we attempt to answer throughout this work:

- is labeled source data sufficient to train a supervised learning model to make accurate predictions on target data?
- does single-source domain adaptation result in the higher accuracy?
- does multi-source domain adaptation result in the higher accuracy as compared to single-source domain adaptation?

In our experiments, we repeatedly take 2012 Sandy Hurricane, 2013 Queensland Floods, 2013 Boston Bombings, 2013 West Texas Explosion, 2013 Oklahoma Tornado and 2013 Alberta Floods tweets and combine them in source-target pairs based on the chronological order of the actual events. Our motivation is as follows: a *source* disaster happens earlier than a *target* disaster and thus, training a classifier in such a way complies more with future real-life applications.

Chapter 2

Related work

In this chapter we discuss some of the previous work that has been done in the field of domain adaptation and review some of the relevant research papers.

Domain Adaptation has been researched in a number of various machine learning applications, for instance, text classification [Dai et al., 2007], bioinformatics [Herndon and Caragea, 2015], cross-domain image retrieval [Huang et al., 2015], multi-task learning [Liu et al., 2015], sentiment classification [Wu and Huang, 2016], among others.

[Dai et al., 2007] propose a novel transfer-learning algorithm for text classification based on an EM-based Naive Bayes classifiers. Their solution is to first estimate the initial probabilities under a distribution D_l of one labeled data set, and then use an EM algorithm to revise the model for a different distribution D_u of the test data which are unlabeled. According to [Dai et al., 2007], the algorithm is very effective in several different pairs of domains, where the distances between the different distributions are measured using the Kullback–Leibler(KL) divergence. In their experiments, they show that the algorithm outperforms the traditional supervised and semi-supervised learning algorithms when the distributions of the training and test sets are increasingly different.

[Herndon and Caragea, 2015] propose an approach for the task of splice site prediction. They use a weighted Nave Bayes classifier, and analyze the three methods for incorporating the target unlabeled data: EM with soft-labels, ST with hard-labels, and also a combination

of EM/ST (with hard-labels for the most confidently labeled instances in the current target unlabeled data, and soft-labels for the other instances). They provide empirical results on splice site prediction indicating that using soft labels only can lead to better classifier compared to the other two ways.

[Huang et al., 2015] propose a Dual Attribute-aware Ranking Network (DARN) for retrieval feature learning. DARN consists of two sub-networks, one for each domain with similar structure. Each of the two domain images are fed into each of the two sub-networks. As [Huang et al., 2015] states, the proposed method is different from previous approaches in that it simultaneously embeds semantic attribute information and visual similarity constraints into the feature learning stage, while modeling the discrepancy of the two domains.

Similarly, [Liu et al., 2015] adopts the deep learning approach, and developes a multi-task DNN for learning representations across multiple tasks. According to the authors, their multi-task DNN approach combines tasks of multiple-domain classification (for query classification) and information retrieval (ranking for web search). gains over strong baselines in a comprehensive set of domain adaptation.

Also, the idea of learning from multiple sources is researched by [Wu and Huang, 2016] in the area of sentiment classification. [Wu and Huang, 2016] propose a new domain adaptation approach which can exploit sentiment knowledge from multiple source domains. They first extract both global and domain-specific sentiment knowledge from the data of multiple source domains using multi-task learning. Then they transfer the knowledge from source domains to target domain with the help of words sentiment polarity relations extracted from the unlabeled target domain data. The authors state that experimental results show the effectiveness of the approach in improving cross-domain sentiment classification performance. However, their approach is not quite transferable to other problems. The reason is that it might be difficult to apply their method to other datasets because we would first need to build a sentiment graph, on which the method heavily relies, and this is not scalable and not trivial.

There has been some research done in the area of disaster management using tweets,

by [Li et al., 2015] and by [Imran et al., 2016], among others.

[Li et al., 2015] study the usefulness of labeled data from a prior source disaster, together with unlabeled data from the current target disaster to learn domain adaptation classifiers for the target. Experimental results suggest that, for some tasks, source data itself can be useful for classifying target data. However, for tasks specific to a particular disaster, domain adaptation approaches that use target unlabeled data in addition to source labeled data are superior.

[Imran et al., 2016] research the performance of the classifiers trained using different combinations of training sets obtained from past disasters. They perform extensive experimentation on real crisis datasets and show that the past labels are useful when both source and target events are of the same type (e.g. both earthquakes). For similar languages, cross-language domain adaptation is useful, however, when for different languages the performance decreases.

In this work, we first look closely at the approach described in [Sun et al., 2016] where they propose a simple, effective, and efficient method for unsupervised domain adaptation called CORrelation ALignment (CORAL), and use it in our single-source domain adaptation setting. CORAL aligns the input feature distributions of the source and target domains by exploring their second-order statistics. The method is frustratingly easy to implement: the only computation involved is recoloring the whitened source features with the covariance of the target domain. Extensive experiments on standard benchmarks demonstrate the superiority of their method over many existing state-of-the-art methods. These results confirm that CORAL is applicable to multiple features types, including highlyperforming deep features, and to different tasks, including computer vision and natural language processing.

Furthermore, we adopt the idea proposed in [Zhang et al., 2015] and use it in our multi-source domain adaptation setting. [Zhang et al., 2015] use causal models to represent the relationship between the features X and class label Y, and consider possible situations where different modules of the causal model change with the domain. In each situation, they investigate what knowledge is appropriate to transfer and find the optimal target-domain hypothesis. They finally focus on the case where Y is the cause for X with changing P_Y

and $P_{X|Y}$, that is, P_Y and $P_{X|Y}$ change independently across domains. Precisely, under appropriate assumptions, the availability of multiple source domains allows a natural way to reconstruct the conditional distribution on the target domain. They propose to model $P_{X|Y}$ (the process to generate effect X from cause Y) on the target domain as a linear mixture of those on source domains, and estimate all involved parameters by matching the target-domain feature distribution. According to [Zhang et al., 2015], experimental results on both synthetic and real-world data verify their theoretical results.

Chapter 3

Data Description

In this chapter, we first present the original dataset in Section 3.1, and then describe the operations performed on the data (i.e. preprocessing) in Section 3.2.

3.1 Original Dataset

In this work, we use the dataset CrisisLexT6 [Olteanu et al., 2014]. The dataset consists of 60,000 tweets[1] posted during 6 crisis events in 2012 and 2013. The 60,000 tweets (10,000 in each disaster) have been labeled by crowdsourcing workers according to relatedness (as *on-topic* or *off-topic*). The *on-topic* tweets are labeled as *1*, and the *off-topic* tweets are labeled as *0*. The amount of tweets per class for each disaster is presented in Table 3.1.

Table 3.1: *Original Dataset*

Crisis	On-topic	Off-topic	Total
2012 Sandy Hurricane	6138	3870	10008
2013 Queensland Floods	5414	4619	10033
2013 Boston Bombings	5648	4364	10012
2013 West Texas Explosion	5246	4760	10006
2013 Oklahoma Tornado	4827	5165	9992
2013 Alberta Floods	5189	4842	10031

[1] A tweet is a short text (up to 140 characters) that users of Twitter can post on Twitter.com

3.2 Data Preprocessing

The tweets are preprocessed before they are used in training, domain adaptation and testing stages. The following cleaning steps have been taken [Li et al., 2015]:

- non-printable, ASCII characters are removed, as they are generally regarded as noise rather than useful information.
- printable HTML entities are converted into their corresponding ASCII equivalents
- URLs, email addresses, and usernames are replaced with a URL/email/username placeholder for each type of entity, respectively, under the assumption that those features could be predictive
- numbers, punctuation signs and hashtags are kept under the assumption that numbers could be indicative of an address, while punctuation/emoticons and hashtags could be indicative of emotions
- RT (i.e., retweet) are removed under the assumptions that such features are not informative for our classification tasks
- duplicate tweets and empty tweets (that have no characters left after the cleaning) are removed from the data sets

The amount of tweets per class for each disaster has reduced and is presented in Table 3.2.

Table 3.2: *Dataset after preprocessing*

Crisis	On-topic	Off-topic	Total
2012 Sandy Hurricane	5261	3752	9013
2013 Queensland Floods	3236	4550	7786
2013 Boston Bombings	4441	4309	8750
2013 West Texas Explosion	4123	4733	8856
2013 Oklahoma Tornado	3209	5049	8258
2013 Alberta Floods	3497	4714	8211

After preprocessing, the source tweets are expressed via target features i.e. via words that occur in the target tweets. The bag-of-words [Mitchell, 1997] representation is used to represent tweets as vectors of features. A sample bag-of-words tweet represenatation is presented in Table 3.3.

Table 3.3: *Example of bag-of-words representation*

Sample Tweet	life	is	real	now	New	York	has	been	real	travelling	to
Life is real now	1	1	1	1	0	0	0	0	0	0	0
Travelling to New York now	0	0	0	1	1	1	0	0	0	1	1
New York has been real	0	0	1	0	1	1	1	1	1	0	0

Chapter 4

Single-source Domain Adaptation Approach

In this chapter, we define the problem of learning from a single source in Section 4.1, and describe the correlation alignment algorithm ("CORAL") proposed in the paper [Sun et al., 2016] in Section 4.2. Then we discuss the results obtained after applying CORAL in Section 4.3.

4.1 Problem Definition

We define our goal as follows: given tweets from a single source domain, train a model to classify tweets from a target domain. However, direct usage of source data may not give good performance, even if it is expressed via target features. So, we attempt to perform some transformations on source data to align its distribution with the target data distribution, assuming that labels for the target data are not available. As a possible solution, we adopt the method described in [Sun et al., 2016], which minimizes the domain shift by aligning the second-order statistics of source and target distributions, without requiring any target labels.

4.2 Correlation Alignment Algorithm

[Sun et al., 2016] present an extremely simple domain adaptation method — CORrelation ALignment (CORAL) — which works by aligning the distributions of the source and target features in an unsupervised manner. They propose to match the distributions by aligning the second-order statistics, namely, the covariance. More concretely, as [Sun et al., 2016] states, CORAL aligns the distributions by re-coloring whitened source features with the covariance of the target distribution. CORAL is simple and efficient, as the only computations it needs are:

- computing covariance statistics in each domain
- applying the whitening and re-coloring linear transformation to the source features.

Then, supervised learning proceeds as usual – training a classifier on the transformed source features.

They describe their method by taking a multi-class classification problem as the running example. Given source-domain training examples $D_S = \{\overrightarrow{x_i}\}$, $\overrightarrow{x} \in \mathbb{R}^D$ with labels $L_S = \{y_i\}$, $y \in \{1, \cdots, L\}$, and target data $D_T = \{\overrightarrow{u_i}\}$, $\overrightarrow{u} \in \mathbb{R}^D$. Here both $\{\overrightarrow{x}\}$ and $\{\overrightarrow{u}\}$ are the D-dimensional feature representations $\varphi(I)$ of input I.

Suppose μ_s, μ_t and C_S, C_T are the feature vector means and covariance matrices. According to [Sun et al., 2016], to minimize the distance between the second-order statistics (covariance) of the source and target features, they apply a linear transformation A to the original source features and use the Frobenius norm as the matrix distance metric:

$$\min_A \|C_{\hat{S}} - C_T\|_F^2 = \min_A \left\|A^T C_S A - C_T\right\|_F^2$$

where C_S is covariance of the transformed source features $D_s A$ and $\|\cdot\|_F^2$ denotes the matrix Frobenius norm. Essentially, the solution lies in finding the matrix A.

After a series of calculations, which are presented in [Sun et al., 2016] in detail, the

optimal solution can be found as:

$$A^* = U_S E = (U_S \Sigma_S^{+\frac{1}{2}} U_S^\top)(U_{T[1:r]} \Sigma_{T[1:r]}^{+\frac{1}{2}} U_{T[1:r]}^\top)$$

This can be interpreted as follows: the first part whitens the source data while the second part re-colors it with the target covariance.

Whitening refers to the process of first de-correlating the data y – its covariance, $\mathbf{E}(yy^\top)$ is now a diagonal matrix, Λ. The diagonal elements (eigenvalues) in Λ may be the same or different. If we make them all the same, then this is called whitening the data [Picard, 2010].

Re-coloring generally refers to the process of transforming a vector of white random variables into a random vector with a specified covariance matrix [Hossain, 2010].

As [Sun et al., 2016] suggests, after CORAL transforms the source features to the target space, a classifier $f_{\vec{w}}$ parametrized by ω can be trained on the adjusted source features and directly applied to target features.

Since correlation alignment changes the features only, it can be applied to any base classifier. In this work, we run experiments using Naive Bayes Classifier.

The first supervised learning method we use is the multinomial Naive Bayes or multinomial NB model, a probabilistic learning method [Manning et al., 2009]. It is used for multinomially distributed data, for instance, in text classification where the data are typically represented as word vector counts. We use it with *counts* representation for the data.

An alternative to the multinomial model is the multivariate Bernoulli model or Bernoulli model. It generates an indicator for each term of the vocabulary, either 1 indicating presence of the term in the document or 0 indicating absence [Manning et al., 2009]. We use it with *0/1* representation for the data.

However, since CORAL changes the values of the data from *0/1* to continuous, we need to use Gaussian Naive Bayes algorithm for classification of such data. The likelihood of the features is assumed to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

The results of the experiments are discussed in Section 4.3.

4.3 Experiments and Results

We setup the experiments as follows:

- use only target features to represent sources
- perform 5-fold cross-validation over target and report the average accuracy over the 5 folds (each source is "aligned" with 3 target unlabeled folds using CORAL, 1 labeled target fold is used for testing, 1 labeled target fold is kept for possible use of labeled target data)
- vary the number of instances in the sources (i.e. 500 instances per class, 1000 instances per class, etc) - smaller datasets are subsets of th larger datasets

4.3.1 Preliminary Results without Domain Adaptation

We present the results obtained when no domain adaptation is performed. The source data is expressed via target features and is described in Table 4.1. Then, two representations are tested: *0/1* and *counts* to determine which representation is more promising i.e. gives better results. Bernoulli Naive Bayes and Multinomial Naive Nayes classifiers are used for *0/1* and *counts* representations respectively.

The target data is divided into five folds for cross-validation [Hastie et al., 2009]. Each fold in turn is used for testing, and the accuracy is recorded in each run. The average results are reported. The number of instances per class in the source data is varied: the results are recorded for source data having 500 instances per class, 1000 instances per class, and 2000 instances per class. All source data is also used as training data resulting in slightly better accuracy for $Source2$ and $Source3$ (0.7888 and 0.7421, respectively, as compared to 0.7810 and 0.7346, respectively, for the number of instances equal to 2000) in Table 4.2. Generally,

balanced training data (i.e. data that has equal number of instances per class) gives better performance.

The results in Table 4.2 and Table 4.3 indicate that *0/1* representation is more efficient. One of the possible explanations might be that it is unlikely that meaningful words are repeatedly used in a single tweet since tweets are relatively short. Thus, keeping the actual counts does not provide additional relevant information.

Table 4.1: *Target and Source datasets*

Target	Source 1	Source 2	Source 3
2013 Alberta Floods	2012 Sandy Hurricane	2013 Queensland Floods	2013 Boston Bombings

Table 4.2: *Accuracy after running Bernoulli Naive Bayes with 0/1 representation for instances*

Sources	500	1000	2000	All
Source 1	0.699183336	0.737669795	0.748874085	0.714163926
Source 2	0.759834769	0.764704356	0.781025693	0.788819876
Source 3	0.710511727	0.716357069	0.734625396	0.742175131

Table 4.3: *Accuracy after running Multinomial Naive Bayes with counts representation for instances*

Sources	500	1000	2000	All
Source 1	0.667883087	0.708560882	0.727195729	0.694068932
Source 2	0.749968752	0.758614519	0.772012294	0.786384119
Source 3	0.676287695	0.697966125	0.715016795	0.720984046

After running the preliminary experiments both with *0/1* representation and *counts* representation using Naive Bayes Classifier [Mitchell, 1997], the results obtained in Table 4.2 have shown that *0/1* representation gives better performance as compared to the *counts* representation results presented in Table 4.3. Consequently, in the further experiments the *0/1* representation is used.

We can see improvement for one pair of source, precisely, 2012 Sandy Hurricane. After applying CORAL, the accuracy increases from 0.74 to 0.77 for a subset of 2000 instances

Table 4.4: *Accuracy after applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.670077908	0.733406627	0.77286521	0.797223237
2013 Queensland Floods	0.666543406	0.645719596	0.684447955	0.657191572
2013 Boston Bombings	0.643529372	0.573255749	0.599684114	0.649713799

per class and from 0.71 to 0.79 for a set that uses all instances. In other cases, the accuracy decreases. One of the possible reasons might be that the initial matrix is very sparse and has a large number of features (precisely, 1334) as compared to the number of instances (1000, 2000 etc). In the original paper [Sun et al., 2016], they reduce the original dimentionaly of the dataset to keep top 400 features based on mutual information [Hastie et al., 2009].

Similarly, we attempt to reduce the dimentionaly of our dataset. We experiment with several dimentionaly reduction techniques. Since we assume that target labeled data is not available, using mutual information for feature selection is not quite applicable.

4.3.2 Feature Selection

Principle Component Analysis

One of the dimentionality reduction methods that does not require labeled data is Principle Component Analysis (PCA) – standard linear principal components are obtained from the eigenvectors of the covariance matrix, and give directions in which the data have maximal variance [Hastie et al., 2009]. Basically, we select k principle components that should describe our data well. One of the recommended methods to choose k is to choose the smallest k for which 99% of variance retained. However, when we choose k in this way, the number of components (i.e. features) retained becomes 1332, which is only 2 features less than the original dimentionality of 1334. Therefore, we decide to not proceed with using PCA on our data.

Variance Threshold

Feature selector that removes all low-variance features.

This feature selection algorithm looks only at the features X, not the desired outputs y, and can thus be used for unsupervised learning [Scikit-Learn, 2016]. Specifically, we have a dataset with boolean features, and we want to remove all features that are either one or zero (on or off), for instance, in more than 99% of the samples. Boolean features are Bernoulli random variables, and the variance of such variables is given by:

$$Var[X] = p(1-p)$$

so we can select features using the threshold equal to $.99 * (1 - .99)$.

In order to select features, we first concatenate source data and target data, which we call *training target unlabeled – tTU*. Second, we transform source data and tTU using the extracted features. Then, we run CORAL on source data and tTU. We fit the classifier with transformed source data after the CORAL stage, and we test on *testing target data*.

Features with a training-set variance lower than the specified threshold are removed. We apply feature selection for every combination of source-target fold, so the number of features varies from 160 to 176. That is a significant reduction (by 88%) compared to the original dimentionaly of the data – 1334.

The value of threshold is varied across the experiments. Precisely, we experiment with $k = 0.95, k = 0.90, k = 0.80$, and we present results in Table 4.7, in Table 4.8, in Table 4.9 respectively. The highest accuracy is obtained when the threshold is equal to 0.99 as described in Table 4.6.

We decide to further check whether feature selection by itself improves performance, without applying CORAL. We run select features based on Varience Threshold equal to 0.99 and then run the Bernoulli Naive Bayes classifier. The results presented in Table 4.5 show that applying CORAL does improve performance. Thus, eliminating features with variance lower than 0.99 and then applying CORAL is more efficient than when CORAL is not applied.

We decide to test the method further on all pairs of source and target disasters described in Table 4.10. The results of applying feature selection followed by CORAL are shown in

Table 4.5: *Accuracy after selecting features via Variance Threshold (0.99) without applying CORAL with Bernoulli Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.669101929	0.701010747	0.715869636	0.651807358
2013 Queensland Floods	0.740105627	0.740835257	0.742905828	0.74838643
2013 Boston Bombings	0.685786895	0.694311377	0.69711321	0.695408269

Table 4.6: *Accuracy after selecting features via Variance Threshold (0.99) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.770675727	0.843014879	0.857263347	0.846181082
2013 Queensland Floods	0.732553416	0.807941416	0.813179302	0.802460296
2013 Boston Bombings	0.712096496	0.712092048	0.744128673	0.791619412

Table 4.12, and the results of applying feature selection without CORAL are presented in Table 4.11.

In addition, we present the number of features retained after applying Variance Threshold to the pair 2013 Boston Bombings – 2013 Alberta Floods when the different amount of the source instances is used in Figure 4.1. Precisely, 1000 number of the source instances means that there are 500 instances of the class *1*, and 500 instances of the class *0*, the same logic applies to 2000 and 4000. 8750 instances means that we use all the instances in the source data, and it becomes imbalanced. The plot gives a better understanding of the effect of Variance Threshold on reducing the feature space. The number of features for other pairs is in a similar range of $159 - 187$, which varies for different pairs and their folds.

Interestingly, when the threshold is set to 0.95, the number of features retained reduces even further, and is presented in Figure 4.2.

We decide to further experiment with this feature selection method, precisely, we now rank features by their variance i.e. calcucating $p(1-p)$ for each feature across all samples, in the descending order. We then take top k features and run the experiments again. The results for $k = 190$ without applying CORAl and with applying CORAL are shown in Table 4.13 and in Table 4.14, respectively, and the results for $k = 170$ with applying CORAL are shown in Table 4.15.

Table 4.7: *Accuracy after selecting features via Variance Threshold (0.95) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.589452614	0.775300967	0.735233297	0.652419188
2013 Queensland Floods	0.837409658	0.81122223	0.837044324	0.698949665
2013 Boston Bombings	0.639519076	0.706117934	0.781025248	0.734260358

Table 4.8: *Accuracy after selecting features via Variance Threshold (0.90) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.703077612	0.681526025	0.639758233	0.592012176
2013 Queensland Floods	0.815486955	0.754715498	0.734633995	0.657777097
2013 Boston Bombings	0.743512617	0.700768106	0.679461385	0.626111366

Truncated SVD aka Latent Semantic Analysis

This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD) [Manning et al., 2009]. Contrary to PCA, this estimator does not center the data before computing the singular value decomposition. This means it can work with scipy.sparse matrices efficiently.

The results on our data show that Truncated SVD does not contribute to better accuracy. We manually choose the number of components k. Concretely, we experiment with $k = 677$, choosing this values as 50% of the original number of features (1334), with $k = 400$ and with $k = 170$, choosing this value as an average number of features obtained after Variance Threshold discussed in 4.3.2 that give the best performance so far.

Mutual Information

The mutual information of two random variables is a natural measure of dependence between the two variables [Hastie et al., 2009], which can be expressed as follows:

$$I(x, y) = \sum_{x,y} P(x, y) \ln \frac{P(x, y)}{P(x)P(y)}$$

Table 4.9: *Accuracy after selecting features via Variance Threshold (0.80) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.561565954	0.590795039	0.585071351	0.517477091
2013 Queensland Floods	0.62684663	0.616980465	0.636341457	0.644623001
2013 Boston Bombings	0.743512617	0.700768106	0.679461385	0.626111366

Table 4.10: *All Pairs of Source-Target Disasters*

Pair	Source Disaster Event	Target Disaster Event
$SH \rightarrow QF$	2012 Sandy Hurricane	2013 Queensland Floods
$SH \rightarrow BB$	2012 Sandy Hurricane	2013 Boston Bombings
$QF \rightarrow BB$	2013 Queensland Floods	2013 Boston Bombings
$SH \rightarrow WT$	2012 Sandy Hurricane	2013 West Texas Explosion
$BB \rightarrow WT$	2013 Boston Bombings	2013 West Texas Explosion
$SH \rightarrow OT$	2012 Sandy Hurricane	2013 Oklahoma Tornado
$QF \rightarrow OT$	2013 Queensland Floods	2013 Oklahoma Tornado
$BB \rightarrow OT$	2013 Boston Bombings	2013 Oklahoma Tornado
$SH \rightarrow AF$	2012 Sandy Hurricane	2013 Alberta Floods
$QF \rightarrow AF$	2013 Queensland Floods	2013 Alberta Floods
$BB \rightarrow AF$	2013 Boston Bombings	2013 Alberta Floods

Our first experiment in this subsection we setup as follows:

- express source via target features
- select top K features from source based on Mutual Information
- transform the source and target validation fold to be expressed via newly obtained top K features
- run CORAL on transformed source and transformed target obtained at the previous step
- fit the classifier with the transformed source obtained at the previous step
- transform the target test fold to be expressed via top K features
- test the classifier on target test fold

Table 4.11: *Accuracy after selecting features via Variance Threshold (0.99) without applying CORAL with Bernoulli Naive Bayes. All pairs*

Pair	500	1000	2000	Total
$SH \rightarrow QF$	0.645647344	0.76355232	0.773312788	0.724891438
$SH \rightarrow BB$	0.694285714	0.7664	0.703314286	0.6864
$QF \rightarrow BB$	0.712685714	0.703314286	0.716342857	0.717485714
$SH \rightarrow WT$	0.682135879	0.735997122	0.714882615	0.738481467
$BB \rightarrow WT$	0.923893032	0.931683902	0.93157129	0.94241173
$SH \rightarrow OT$	0.763017791	0.768710137	0.795108695	0.762413197
$QF \rightarrow OT$	0.796802144	0.801283322	0.815331232	0.815209727
$BB \rightarrow OT$	0.796561919	0.791354935	0.806612765	0.808791942
$SH \rightarrow AF$	0.669101929	0.701010747	0.715869636	0.651807358
$QF \rightarrow AF$	0.740105627	0.740835257	0.742905828	0.74838643
$BB \rightarrow AF$	0.685786895	0.694311377	0.69711321	0.695408269

Table 4.12: *Accuracy after selecting features via Variance Threshold (0.99) and applying CORAL with Gaussian Naive Bayes. All pairs*

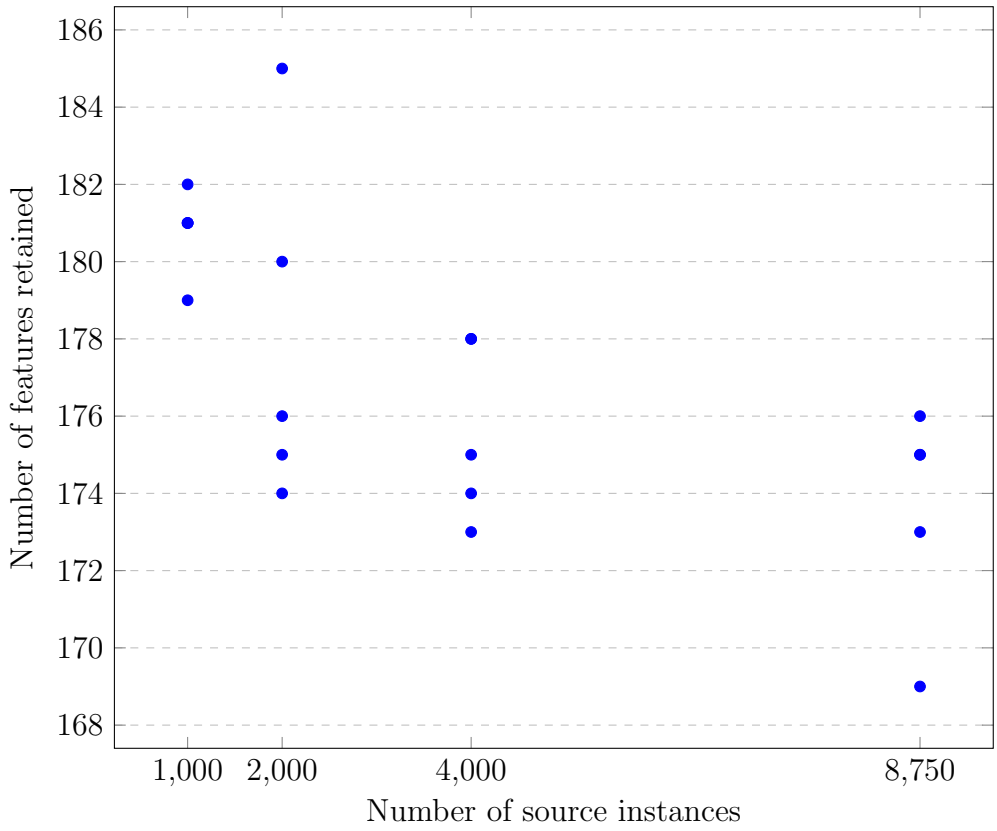
Pair	500	1000	2000	Total
$SH \rightarrow QF$	0.751221161	0.851656398	0.852042661	0.838684627
$SH \rightarrow BB$	0.789714286	0.764114286	0.827314286	0.766971429
$QF \rightarrow BB$	0.834057143	0.819657143	0.804571429	0.68
$SH \rightarrow WT$	0.802055629	0.738821915	0.674232334	0.836269315
$BB \rightarrow WT$	0.888098892	0.945347159	0.94568627	0.949412532
$SH \rightarrow OT$	0.85359559	0.858683049	0.857835445	0.753334467
$QF \rightarrow OT$	0.82743645	0.867645187	0.873215221	0.815452151
$BB \rightarrow OT$	0.792924312	0.853962013	0.827806759	0.823568371
$SH \rightarrow AF$	0.770675727	0.843014879	0.857263347	0.846181082
$QF \rightarrow AF$	0.732553416	0.807941416	0.813179302	0.802460296
$BB \rightarrow AF$	0.712096496	0.712092048	0.744128673	0.791619412

We select $k = 300$ and present the results in Table 4.19.

Next, we increase the parameter k to be equal to 400 to retain more features, and presumably, improve performance. The results are shown in Table 4.20. We conclude that keeping more features improve the accuracy. However, the way we apply the Mutual Information selection method on source might not be optimal.

We combine source and target data disregarding original source labels and assigning label 0 to source samples, and label 1 to target samples. Then we select top K features based on

Figure 4.1: *Number of features retained after applying Variance Threshold (0.99)* Source – 2013 Boston Bombings, Target – 2013 Alberta Floods



mutual information with the labels.

Figure 4.2: *Number of features retained after applying Variance Threshold (0.95)* Source – 2013 Boston Bombings, Target – 2013 Alberta Floods

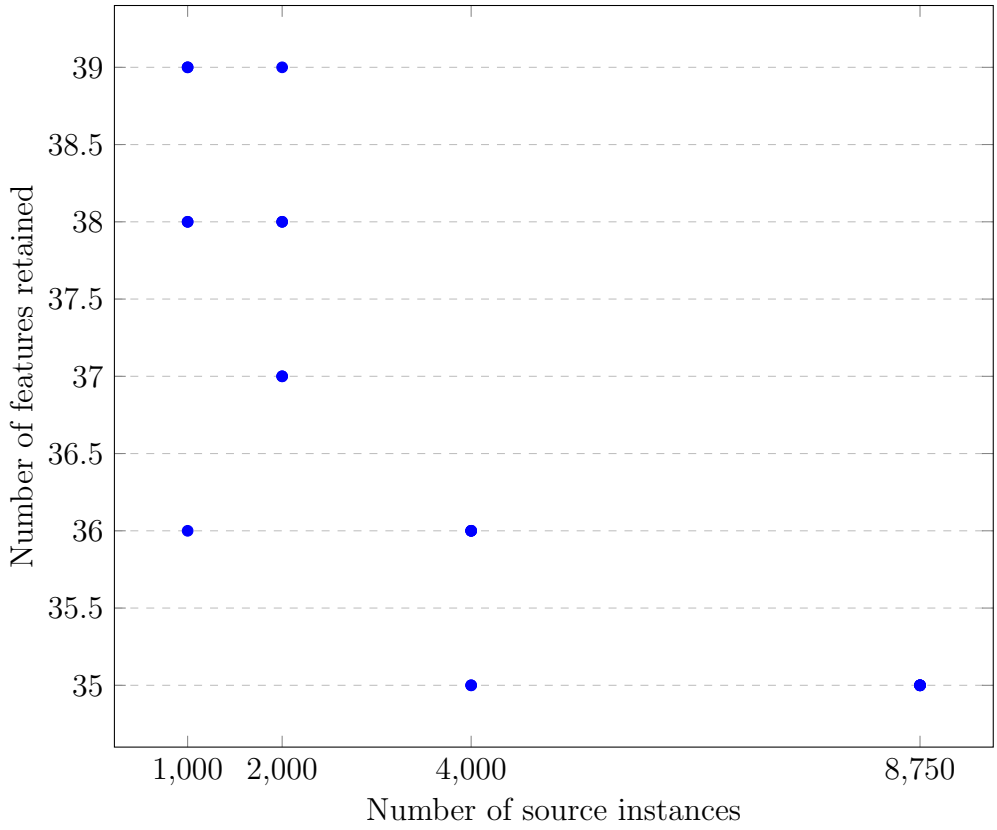


Table 4.13: *Accuracy after ranking features by variance in the descending order, and selecting top k features (k=190), without applying CORAL with Bernoulli Naive Bayes. All pairs*

Pair	500	1000	2000	Total
$SH \rightarrow QF$	0.650399414	0.77138584	0.78179063	0.730799495
$SH \rightarrow BB$	0.693485714	0.766171429	0.7048	0.692342857
$QF \rightarrow BB$	0.709714286	0.710057143	0.718628571	0.725714286
$SH \rightarrow WT$	0.681909635	0.735659031	0.7162381	0.73915905
$BB \rightarrow WT$	0.924231696	0.931345237	0.932022884	0.942976255
$SH \rightarrow OT$	0.766046699	0.775854974	0.804795391	0.778155509
$QF \rightarrow OT$	0.801041925	0.804068706	0.818116396	0.823080443
$BB \rightarrow OT$	0.798257348	0.790748508	0.807339011	0.81714736
$SH \rightarrow AF$	0.67555547	0.710753775	0.718305616	0.658506134
$QF \rightarrow AF$	0.744489856	0.742174641	0.746437142	0.752405103
$BB \rightarrow AF$	0.694069181	0.70393327	0.702228626	0.693703105

Figure 4.3: *Number of features retained after applying Variance Threshold (0.90)* Source – 2013 Boston Bombings, Target – 2013 Alberta Floods

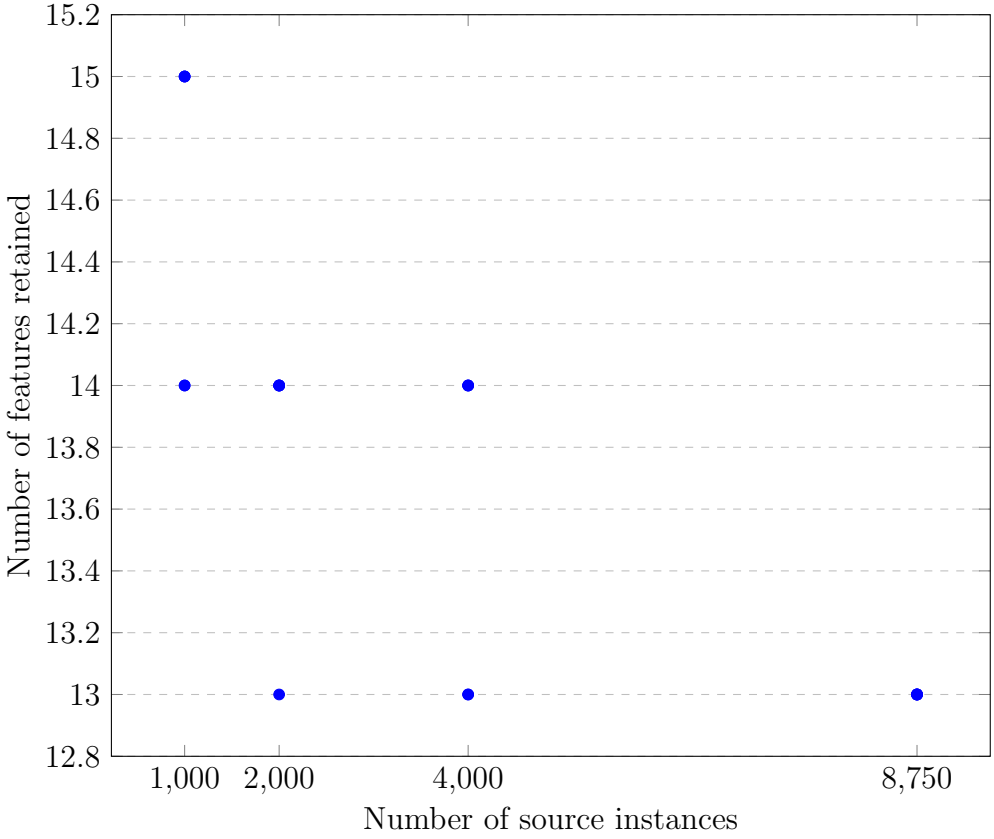


Table 4.14: *Accuracy after ranking features by variance in the descending order, selecting top k features (k=190), and applying CORAL with Gaussian Naive Bayes. All pairs*

Pair	500	1000	2000	Total
$SH \rightarrow QF$	0.78114606	0.858591907	0.860391309	0.830336061
$SH \rightarrow BB$	0.793714286	0.7704	0.818742857	0.766857143
$QF \rightarrow BB$	0.753828571	0.791085714	0.810285714	0.670628571
$SH \rightarrow WT$	0.791216081	0.741531547	0.686653802	0.832543181
$BB \rightarrow WT$	0.883921163	0.945686079	0.948508832	0.95076769
$SH \rightarrow OT$	0.850326605	0.866917695	0.860378551	0.770045156
$QF \rightarrow OT$	0.806971048	0.86643248	0.867402763	0.827076627
$BB \rightarrow OT$	0.792201513	0.851418247	0.81460484	0.827200625
$SH \rightarrow AF$	0.771767948	0.834732001	0.863352517	0.853122574
$QF \rightarrow AF$	0.722325697	0.806237216	0.816832789	0.759592573
$BB \rightarrow AF$	0.652545809	0.705151075	0.70296389	0.771159157

Table 4.15: *Accuracy after ranking features by variance in the descending order, selecting top k features (k=170), and applying CORAL with Gaussian Naive Bayes. All pairs*

Pair	500	1000	2000	Total
$SH \rightarrow QF$	0.744414351	0.842666149	0.854740156	0.83958338
$SH \rightarrow BB$	0.789714286	0.763085714	0.827428571	0.765942857
$QF \rightarrow BB$	0.812914286	0.8128	0.808457143	0.667657143
$SH \rightarrow WT$	0.80826528	0.739837334	0.676491072	0.834688224
$BB \rightarrow WT$	0.886856528	0.944217854	0.945460409	0.949525462
$SH \rightarrow OT$	0.852627361	0.858683049	0.858319706	0.75309219
$QF \rightarrow OT$	0.835914399	0.866071557	0.868976759	0.823080516
$BB \rightarrow OT$	0.793046184	0.84390794	0.824658106	0.827927604
$SH \rightarrow AF$	0.76263675	0.8427712	0.858968584	0.845937625
$QF \rightarrow AF$	0.73230981	0.807819613	0.814884317	0.791254449
$BB \rightarrow AF$	0.718186778	0.711848591	0.738403725	0.788696593

Table 4.16: *Accuracy after selecting features via Truncated SVD (k=677) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.610885735	0.61076534	0.631833052	0.614783494
2013 Queensland Floods	0.644013172	0.648520761	0.651077431	0.646450264
2013 Boston Bombings	0.635488616	0.58519041	0.606746297	0.609668153

Table 4.17: *Accuracy after selecting features via Truncated SVD (k=400) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.629033148	0.630251842	0.64462226	0.634392095
2013 Queensland Floods	0.656436971	0.617708835	0.626234503	0.606870175
2013 Boston Bombings	0.646330092	0.56668048	0.578736203	0.582876604

Table 4.18: *Accuracy after selecting features via Truncated SVD (k=170) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.610885735	0.61076534	0.631833052	0.614783494
2013 Queensland Floods	0.644013172	0.648520761	0.651077431	0.646450264
2013 Boston Bombings	0.635488616	0.58519041	0.606746297	0.609668153

Table 4.19: *Accuracy after selecting features based on Mutual Information in Source (k=300) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.668858769	0.710877135	0.648762884	0.698939879
2013 Queensland Floods	0.704055592	0.742418098	0.708438635	0.63378575
2013 Boston Bombings	0.46681585	0.6857886	0.500670619	0.690170383

Table 4.20: *Accuracy after selecting features based on Mutual Information in Source (k=400) and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.670563117	0.725370023	0.751186186	0.7265859
2013 Queensland Floods	0.721959474	0.680918495	0.632077177	0.63512432
2013 Boston Bombings	0.538304904	0.697602422	0.526854118	0.720617346

Table 4.21: *Accuracy after selecting features based on Mutual Information in Source (k=300) combining Source and Target, and applying CORAL with Gaussian Naive Bayes*

Source Disaster Event	500	1000	2000	Total
2012 Sandy Hurricane	0.0.725121673	0.812701655	0.729157619	0.824256229
2013 Queensland Floods	0.747290799	0.510790991	0.75946773	0.506013627
2013 Boston Bombings	0.663243836	0.441239214	0.29570236	0.382660725

Chapter 5

Multi-source Domain Adaptation Approach

In this chapter, we first define the problem of learning from multiple sources in Section 5.1, and then describe the multi-source domain adaptation algorithm ("MDA") proposed in the paper [Zhang et al., 2015] in Section 5.2. Finally, we discuss the results obtained after applying MDA in Section 5.3.

5.1 Problem Definition

We define our goal as follows: given tweets from several source domains, train a model to classify tweets from a target domain. The general intuition is that more data should improve performance. Yet, adding more source data, even when expressed via target features, may not necessarily contribute to a higher classification accuracy. In addition, labeled target data, again, is not available. Thus, we explore one of the methods presented in [Zhang et al., 2015], specificaly, we adopt the idea of modeling the target domain as a linear mixture of the source domains.

5.2 Multi-source Domain Adaptation Algorithm

5.3 Experiments and Results

We present the results obtained when no domain adaptation is performed in Table 5.2. The source data is expressed via target features and is described in Table 5.1. The binary representation (i.e. *0/1*) for both source and target data is used. The source data from different domains is merged together and treated as a whole.

The target data is divided into five folds for cross-validation [Hastie et al., 2009]. Each fold in turn is used for testing, and the accuracy is recorded in each run. The average results are reported. The number of instances per class in the source data is varied: the results are recorded for source data having 500 instances per class, 1000 instances per class, and 2000 instances per class. All source data is also used as training data.

As we can see, using more source data generally improves the performance of the classifier.

Table 5.1: *Pairs of Mutli-source–Target Disasters*

Pair	Source Disaster Event	Target Disaster Event
$SH \rightarrow BB$	2012 Sandy Hurricane	2013 Boston Bombings
$QF \rightarrow BB$	2013 Queensland Floods	2013 Boston Bombings
$SH \rightarrow WT$	2012 Sandy Hurricane	2013 West Texas Explosion
$BB \rightarrow WT$	2013 Boston Bombings	2013 West Texas Explosion
$SH \rightarrow OT$	2012 Sandy Hurricane	2013 Oklahoma Tornado
$QF \rightarrow OT$	2013 Queensland Floods	2013 Oklahoma Tornado
$BB \rightarrow OT$	2013 Boston Bombings	2013 Oklahoma Tornado
$SH \rightarrow AF$	2012 Sandy Hurricane	2013 Alberta Floods
$QF \rightarrow AF$	2013 Queensland Floods	2013 Alberta Floods
$BB \rightarrow AF$	2013 Boston Bombings	2013 Alberta Floods

We first choose a pair of $SH, QF, BB \rightarrow AF$ to experiment with the idea proposed in [Zhang et al., 2015]. The results are presented in Table 5.3. The columns 500, 1000, 2000 and $Total$ mean that 500, 1000, 2000 and all samples per class per source are taken, respectively, and weights are obtained for each of them separately. The weights are presented in Table 5.4, which can be interpreted as follows: the weights for a specific class (0 "Neg"

Table 5.2: *Accuracy after running Bernoulli Naive Bayes on multi-source data when no domain adaptation is performed*

Pair	500	1000	2000	Total
$SH, QF \rightarrow BB$	0.615428571	0.735657143	0.596	0.688685714
$SH, BB \rightarrow WT$	0.850383658	0.877821702	0.90255107	0.930555488
$SH, QF, BB \rightarrow OT$	0.82550285	0.845725901	0.84960058	0.826957615
$SH, QF, BB \rightarrow AF$	0.72804768	0.752038286	0.768482834	0.775057287

or 1 "Pos") across the sources should sum up to 1. The more one source is similar to the target, the more weight it receives. Also, when the source data is balanced i.e. 500, 1000, 2000 instances per class per source are used, the weights do not differ much, for example, the weights for the negative class when 500 intances are used, are almost identical across all the three sources: 0.33779, 0.33601 and 0.3262 for SH, QF and BB, respectively. One of the possible interpretations might be that since the data across sources is balanced, they contribute the equal amount of information for the target.

Table 5.3: *Accuracy after running Gaussian Naive Bayes on multi-source data when domain adaptation is performed*

Pair	500	1000	2000	Total
$SH, QF, BB \rightarrow AF$	0.705416920268	0.753499695679	0.723067559343	0.709068776628

Table 5.4: *Weights obtained for $SH, QF, BB \rightarrow AF$*

Source	Class	500	1000	2000	Total
SH	Neg	0.33779	0.33864	0.35149	0.38421
	Pos	0.34725	0.36871	0.37865	0.38766
QF	Neg	0.33601	0.34244	0.34336	0.38544
	Pos	0.30849	0.25878	0.23889	0.15128
BB	Neg	0.3262	0.31891	0.30515	0.23035
	Pos	0.34426	0.37252	0.38246	0.46105

We decide to further experiment with multi-source domain adaptation. Precisely, motivated by the improved results discussed in Chapter 4, we apply the same logic to the multi-source case: we first select features based on Variance Threshold, and then run CORAL.

The transformed data is used in training of the Gaussian Naive Bayes classifier. The results are presented in Table and in Table .

Table 5.5: *Accuracy after running Gaussian Naive Bayes on multi-source data after applying Variance Threshold (0.99) and CORAL*

Pair	500	1000	2000	Total
$SH, QF \rightarrow BB$	0.811885714	0.7952	0.783542857	0.795771429
$SH, BB \rightarrow WT$	0.924119594	0.943992949	0.94241173	0.932474734
$SH, QF, BB \rightarrow OT$	0.868976026	0.862679013	0.820055275	0.840034068
$SH, QF, BB \rightarrow AF$	0.862623332	0.866762695	0.879185605 v0.857628755	

Chapter 6

Conclusions

In this chapter, we discuss the overall findings and make conclusions based on the results obtained in all of the experiments.

Chapter 7

Future Work

In this chapter, we discuss the possible improvements and limitations of the approches used in the work.

Bibliography

Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culott. Tweedr: Mining twitter to inform disaster response. In *Proceedings of 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014)*, pages 354–358, 2014.

W. Dai, G. Xue, Q. Yang, and Y. Yu. Transferring nave bayes classifiers for text classification. In *In Proceedings of the AAAI 2007 Conference on Artificial Intelligence, Vancouver, British Columbia, Canada*, pages 540–545, 2007.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, 2nd ed.* Springer, 2009.

N. Herndon and D. Caragea. An evaluation of self-training styles for domain adaptation on the task of splice site prediction. In *In Proceedings of the 2015 International Symposium on Network Enabled Health Informatics, Biomedicine and Bioinformatics (HI-BI-BI 2015), Paris, France*, pages 1042–1047, 2015.

Miliha Hossain. Whitening and coloring transforms for multivariate gaussian random variables, 2010. URL https://www.projectrhea.org/rhea/index.php/ECE662_Whitening_and_Coloring_Transforms_S14_MH.

Junshi Huang, Rogerio Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV '15 Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1062–1070, 2015.

M. Imran, P. Mitra, and J. Srivastava. Cross-language domain adaptation for classifying crisis-related short messages. In *In Proceedings of the ISCRAM 2016 Conference Rio de Janeiro, Brazil*, 2016.

Mahima Kaul. Twitter for crisis and disaster relief, 2016. URL https://blog.twitter.com/2016/twitter-for-crisis-and-disaster-relief-in.

Ron Kohavi and Foster Provost. *On Applied Research in Machine Learning. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*. 1998. URL http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html.

Hongmin Li, Nicolais Guevara, Nic Herndon, Doina Caragea, Kishore Nepali, Cornelia Caragea, Anna Squicciarini, and Andrea Tapia. Twitter mining for disaster response: A domain adaptation approach. In *Proceedings of the ISCRAM 2015 Conference*, 2015.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, 2015 Association for Computational Linguistics*, pages 912–921, 2015.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2009.

Dina Fine Maron. How social media is changing disaster response, 2013. URL https://www.scientificamerican.com/article/how-social-media-is-changing-disaster-response/.

Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: Can we trust what we rt? In *In Proceedings of the First Workshop on Social Media Analytics(SOMA '10)*, pages 71–79, 2010.

Tom Mitchell. *Machine learning*. McGraw-Hill Science/Engineering/Math, 1997.

A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises, 2014. URL http://crisislex.org/data-collections.html#CrisisLexT6.

Rosalind W. Picard. Decorrelating and then whitening data, 2010. URL http://courses.media.mit.edu/2010fall/mas622j/whiten.pdf.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach, 3rd ed.* Pearson, 2009.

T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *In Proceedings of the 19th International Conference on World Wide Web (WWW 2010)*, pages 851–860, 2010.

Scikit-Learn. Variance threshold, 2016. URL http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html#sklearn.feature_selection.VarianceThreshold.

Kate Starbird, Leysia Palen, Amanda Hughes, and Sarah Vieweg. Chatter on the red: What hazards threat reveals about the social life of microblogged information. In *In Proceedings of the ACM 2008 Conference on Computer supported cooperative work (CSCW 2010)*, pages 241–250, 2010.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2058–2065, 2016.

T. Terpstra. Towards a realtime twitter analysis during crises for operational crisis management. In *In Proceedings of 9th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2012)*, pages 1–9, 2012.

Fangzhao Wu and Yongfeng Huang. Sentiment domain adaptation with multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016 Association for Computational Linguistics*, pages 301–310, 2016.

Kun Zhang, Mingming Gong, and Bernhard Scholkopf. Multi-source domain adaptation: A causal view. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3150–3157, 2015.