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

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 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 introduce the basic terminology in Section 1.1 in the field of Machine Learning that we use thoughout this document, and we state the main problem addressed in Section 1.2.

We discuss motivation for domain adaptation in Section 1.3 and we also give high-level overview of the approaches used in this work.

Additionally, we describe the data used in the experiments in Section 1.4, as well as the operations performed on the data (e.g. cleaning).

We also discuss the preliminary results in Section 1.5, which are obtained on the data when no domain adaptation is performed.

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 [Stuart Russell, 2009].

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].

Three major types of learning are distinguished: supervised, unsupervised, and reinforcement learning.

In supervised learning the agent is given a training set of N example 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 $y = f(x)$, and it should discover a function h that approximates the true function f [Stuart Russell, 2009].

In unsupervised learning the agent learns patterns in the input even though no explicit feedback is supplied [Stuart Russell, 2009].

In reinforcement learning the agent learns from a series of reinforcementsrewards or punishments [Stuart Russell, 2009].

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

1.2 Problem Definition

We define the task as follows: given source tweets only, train a model to classify target tweets as *relevant/not relevant* i.e. *about disaster/not about disaster*.

One of the key challenges is adapting the classifier to perform well on a target unlabeled domain. This process is called Domain Adaptation and is discussed in Section 1.3.

In my experiments, we repeatedly take 2012 Sandy Hurricane, 2013 Boston Bombings, 2013 Queensland Floods as source tweets and train the classifier on this data, and we take 2013 Alberta Floods tweets as target tweets and test the classifier on it.

1.3 Motivation for Domain Adaptation

In supervised machine learning, the general assumption is that both training and testing data come from the same distribution.

However, in real-world classification problems this assumption rarely holds true. 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. The process of adapting a classifier to make predictions on an unseen domain where labeled data is unavailable is called Domain Adaptation.

1.4 Data description

In this work, we use the dataset CrisisLexT6 [A. Olteanu, 2014]. The dataset consists of 60K tweets[1] posted during 6 crisis events in 2012 and 2013. 60,000 tweets (10,000 in each collection) were 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*.

we select 2013 Alberta Floods tweets as *target* and 2012 Sandy Hurricane, 2013 Boston Bombings, 2013 Queensland Floods tweets as *source*, which are described in detail in Table 1.1 and in Table 1.2.

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

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

- duplicate tweets and empty tweets (that have no characters left after the cleaning) are removed from the data sets

After preprocessing, the source tweets are expressed via target features i.e. via words that occur in the target tweets. The bag-of-words [Wikipedia, a] representation is used to represent tweets as vectors of features.

After running the preliminary experiments discussed in Section 1.5 both with *0/1* representation and *counts* representation using Naive Bayes Classifier [Learn, 2016a], the results obtained in Table 1.3 have shown that *0/1* representation gives better performance as compared to the *counts* representation results presented in Table 1.4. Consequently, in the further experiments the *0/1* representation is used.

Table 1.1: *Target and Source datasets*

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

Table 1.2: *Selected disaster events from CrisisLexT6 dataset*

Disaster Event	On-topic	Off-topic	Total
2013 Alberta Floods	3497	4714	8211
2012 Sandy Hurricane	5261	3752	9013
2013 Queensland Floods	3236	4550	7786
2013 Boston Bombings	4441	4309	8211

1.5 Preliminary results without domain adaptation

We present the results obtained when no domain adaptation is performed. The source data is expressed via target features. 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 to classify target data respectively.

The target data is divided into five folds for cross-validation [Trevor Hastie, 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 1.3. Generally, balanced training data (i.e. data that has equal number of instances per class) gives better performance.

The results in Table 1.3 and Table 1.4 indicate that *0/1* representation is more efficient. One of the possible explanations might be that it is unlikely that 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 1.3: *Accuracy after running Bernoulli Naive Bayes. 0/1 representation of features*

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 1.4: *Accuracy after running Multinomial Naive Bayes. Counts representation of features*

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

Chapter 2

Related work

In this chapter I 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 the number of various machine learning applications.

[Junshi Huang, 2015] address the problem of cross-domain image retrieval by taking clothing products as a concrete use case. They define their task as follows: given an offline clothing image from the street domain, the goal is to retrieve the same or similar clothing items from a large-scale gallery of professional online shopping images. They 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.

According to [Junshi Huang, 2015], the retrieval features from DARN have several advantages compared with the deep features of other proposed networks:

- by using the dual-structure network, the model can handle the cross-domain problem more appropriately
- in each sub-network, the scenario-specific semantic representation of clothing is elaborately captured by leveraging the tree-structure layers

- based on the semantic representation, the visual similarity constraint enables more effective feature learning for the retrieval problem

To analyze the retrieval performance of deep features, [Junshi Huang, 2015] compares pre-trained networks including AlexNet (pretrained CNN) and pre-trained NIN. They show that the attribute-guided learning is a key factor for retrieval accuracy improvement. They evaluate each individual component of the proposed approach. The authors denote the overall solution as Dual Attribute-aware Rank-ing Network (DARN), the solution without dual structure as Attribute-aware Ranking Network (ARN), the solution without dual structure and the ranking loss function as Attribute-aware Network (AN).

Compared with a single model, the dual-structure network greatly improves the retrieval performance, i.e., the top-20 retrieval accuracy of DARN improves 9.9% when compared with ARN.

As [Junshi Huang, 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. [Junshi Huang, 2015] demonstrates the approach in a practical real-world clothing retrieval application, showing substantial improvement over other baselines.

Similarly, [Xiaodong Liu and yi Wang, 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), and demonstrates significant gains over strong baselines in a comprehensive set of domain adaptation.

The multi-task model combines classification and ranking tasks: query classification is used as the classification task and web search is used as the ranking task. [Xiaodong Liu and yi Wang, 2015] writes that the proposed model maps any arbitrary queries Q or documents D into fixed low dimensional vector representations using DNNs. The input is either a query or a document, initially represented as bag-of-words. It goes through a number of layers in the proposed neural network: l_1 - the word hash layer; l_2 - the semantic representation layer;

l_3 - the task-specific representation layer. Thus, the lower layers are shared across different tasks, whereas the top layers represent task-specific outputs.

According to the results presented in [Xiaodong Liu and yi Wang, 2015], the Multi-Task-DNN robustly outperforms strong baselines across all web search and query classification tasks.

Also, the idea of learning from multiple sources is researched by [Fangzhao Wu, 2016] in the area of sentiment classification.

In this paper, the authors 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 them to target domain with the help of words sentiment polarity relations extracted from the unlabeled target domain data. They manually create two relation extraction rules for opposite sentiment polarity, and two relation extraction rules for coherent sentiment polarity, and build a sentiment graph. Each source domain is decomposed into two components:

- the global sentiment model is shared by all source domains and is trained in these domains simultaneously
- the domain-specific sentiment model is trained on the labeled data within one source domain and is used to capture the specific sentiment knowledge of this domain

Furthermore, they mention the domain similary measure based on term distribution, and propose their own domain similarity measure based on similarity between sentiment graphs. Thus, the similarities between target domain and different source domains are also incorporated into the adaptation process.

The authers state that experimental results on benchmark dataset 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.

Chapter 3

Single-source Domain Adaptation Approach

In this chapter I define the problem of learning from a single source in Section 3.1, and describe the correlation alignment algorithm ("CORAL") proposed in the paper [Baochen Sun, 2016] in Section 3.2. Then I discuss the results obtained after applying CORAL in Section 3.3.

3.1 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 [Baochen Sun, 2016], which minimizes domain shift by aligning the second-order statistics of source and target distributions, without requiring any target labels.

3.2 Correlation Alignment Algorithm

[Baochen Sun, 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.We propose to match the distributions by aligning the second-order statistics, namely, the covariance. More concretely, as [Baochen Sun, 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 usualtraining 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 [Baochen Sun, 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 the paper in detail, the optimal solution can be found as:

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

The first part whitens the source data while the second part re-colors it with the target covariance.

As [Baochen Sun, 2016] suggests, after CORAL transforms the source features to the target space, a classifier $f_{\overrightarrow{w}}$ parametrized by ω can be trained on the adjusted source features and directly applied to target features. For a linear classifier $f_{\overrightarrow{w}}(I) = \overrightarrow{w}^T \phi(I)$, we can apply an equivalent transformation to the parameter vector w instead of the features u. This results in added efficiency when the number of classifiers is small but the number and dimensionality of target examples is very high.

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 results of the experiments are discussed in Section 3.3.

3.3 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 target fold is used for testing, 1 target fold is kept as an option for future parameter tuning)
- 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

We can see improvement for one pair of source, precisely, 2012 Sandy Hurricane. After applying CORAL, the accuracy increases from 0.748874085 to 0.77286521 for a subset of 2000

Table 3.1: *Accuracy after applying CORAL. 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

instances per class and from 0.714163926 to 0.797223237 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 significant number of features (precisely, 1334) as compared to the number of instances ($1000, 2000$ etc). In the original paper [Baochen Sun, 2016], they reduce the original dimentionaly of the dataset to keep top 400 features based on mutual information [Wikipedia, b].

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 (although [Baochen Sun, 2016] use the target labeled data, we consider it to be "cheating").

3.3.1 Principle Component Analysis

One of the dimentionality reduction methods that does not require labeled data is Principle Component Analysis (PCA) [Wikipedia, c] where 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.

3.3.2 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 [Learn, 2016c].

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 3.4, in Table 3.5, in Table 3.6 respectively. The highest accuracy is obtained when the threshold is equal to 0.99 as described in Table 3.2.

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 anf then run the Bernoulli Naive Bayes classifier. The results presented in Table 3.3 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.

Table 3.2: *Accuracy after selecting features via Variance Threshold (0.99) and applying CORAL. 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

We decide to test the method further on the following pair of disasters presented in Table 3.7.

Table 3.3: *Accuracy after selecting features via Variance Threshold (0.99) without applying CORAL. 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 3.4: *Accuracy after selecting features via Variance Threshold (0.95) and applying CORAL. 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

3.3.3 Truncated SVD aka Latent Semantic Analysis

This transformer performs linear dimensionality reduction by means of truncated singular value decomposition (SVD). 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 [Learn, 2016b].

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 3.3.2 that give the best performance so far.

3.3.4 Mutual Information

In probability theory and information theory, the mutual information (MI) of two random variables is a measure of the mutual dependence between the two variables. More specifically, it quantifies the "amount of information" (in units such as bits) obtained about one random variable, through the other random variable [Wikipedia, b].

Our first experiment in this subsection we setup as follows:

Table 3.5: *Accuracy after selecting features via Variance Threshold (0.90) and applying CORAL. 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

Table 3.6: *Accuracy after selecting features via Variance Threshold (0.80) and applying CORAL. 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

- 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

We select $k = 300$ and present the results in Table 3.12.

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 3.13. 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.

Table 3.7: *Additional Pairs os Source-Target Disasters*

Pair	Source Disaster Event	Target Disaster Event
SH-¿QF	2012 Sandy Hurricane	2013 Queensland Floods
SH-¿BB	2012 Sandy Hurricane	2013 Boston Bombings
QF-¿BB	2013 Queensland Floods	2013 Boston Bombings
SH-¿WT	2012 Sandy Hurricane	2013 West Texas Explosion
BB-¿WT	2013 Boston Bombings	2013 West Texas Explosion
SH-¿OT	2012 Sandy Hurricane	2013 Oklahoma Tornado
QF-¿OT	2013 Queensland Floods	2013 Oklahoma Tornado
BB-¿OT	2013 Boston Bombings	2013 Oklahoma Tornado

Table 3.8: *Accuracy after selecting features via Variance Threshold (0.99) and applying CORAL. Gaussian Naive Bayes*

Pair	500	1000	2000	Total
SH-¿QF	0.751221161	0.851656398	0.852042661	0.838684627
SH-¿BB	0.789714286	0.764114286	0.827314286	0.766971429
QF-¿BB	0.834057143	0.819657143	0.804571429	0.68
SH-¿WT	0.802055629	0.738821915	0.674232334	0.836269315
BB-¿WT	0.888098892	0.945347159	0.94568627	0.949412532
SH-¿OT	0.85359559	0.858683049	0.857835445	0.753334467
QF-¿OT	0.82743645	0.867645187	0.873215221	0.815452151
BB-¿OT	0.792924312	0.853962013	0.827806759	0.823568371

Table 3.9: *Accuracy after selecting features via Truncated SVD (k=677) and applying CORAL. 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 3.10: *Accuracy after selecting features via Truncated SVD (k=400) and applying CORAL. 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 3.11: *Accuracy after selecting features via Truncated SVD (k=170) and applying CORAL. 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 3.12: *Accuracy after selecting features based on Mutual Information in Source (k=300) and applying CORAL. 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 3.13: *Accuracy after selecting features based on Mutual Information in Source (k=400) and applying CORAL. 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

Chapter 4

Multi-source Domain Adaptation Approach

In this chapter I define the problem of learning from multiple sources in Section 4.1, and describe the multi-source domain adaptation algorithm ("MDA") proposed in the paper [Kun Zhang, 2015] in Section 4.2. Then I discuss the results obtained after applying MDA in Section 4.3.

4.1 Problem Definition

4.2 Multi-source Domain Adaptation Algorithm

4.3 Experiments and Results

Chapter 5

Conclusions

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

Chapter 6

Future Work

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

Bibliography

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

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

Yongfeng Huang Fangzhao Wu. 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.

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

Qiang Chen Shuicheng Yan Junshi Huang, Rogerio Feris. 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.

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

Scikit Learn. Naive bayes, 2016a. URL http://scikit-learn.org/stable/modules/naive_bayes.html.

Scikit Learn. Truncated svd, 2016b. URL http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html.

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

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

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

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

Wikipedia. Bag of words model, a. URL https://en.wikipedia.org/wiki/Bag-of-words_model.

Wikipedia. Mutual information, b. URL https://en.wikipedia.org/wiki/Mutual_information.

Wikipedia. Principal component analysis, c. URL https://en.wikipedia.org/wiki/Principal_component_analysis.

Xiaodong He Li Deng Kevin Duh Xiaodong Liu, Jianfeng Gao 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.