International Journal of Information Systems for Crisis Response and Management
Volume 10 • Issue 4 • October-December 2018

# Domain Adaptation for Crisis Data Using Correlation Alignment and Self-Training

Hongmin Li, Kansas State University, Manhattan, USA

Oleksandra Sopova, Kansas State University, Manhattan, USA

Doina Caragea, Kansas State University, Manhattan, USA

Cornelia Caragea, University of Illinois at Chicago, Chicago, USA

## ABSTRACT

Domain adaptation methods have been introduced for auto-filtering disaster tweets to address the issue of lacking labeled data for an emerging disaster. In this article, the authors present and compare two simple, yet effective approaches for the task of classifying disaster-related tweets. The first approach leverages the unlabeled target disaster data to align the source disaster distribution to the target distribution, and, subsequently, learns a supervised classifier from the modified source data. The second approach uses the strategy of self-training to iteratively label the available unlabeled target data, and then builds a classifier as a weighted combination of source and target-specific classifiers. Experimental results using Naïve Bayes as the base classifier show that both approaches generally improve performance as compared to baseline. Overall, the self-training approach gives better results than the alignment-based approach. Furthermore, combining correlation alignment with self-training leads to better result, but the results of self-training are still better.

## KEYWORDS

Correlation Alignment, Disaster Tweets, Domain Adaptation, Self-Training, Twitter Classification

## INTRODUCTION

From user groups, online forums, to Facebook, Twitter, Instagram, YouTube, social media platforms have become ubiquitous. The use of social media is particularly prevalent during emergencies. For instance, the Federal Emergency Management Agency (FEMA) wrote in its 2013 National Preparedness report (Maron, 2013) that during and immediately following Hurricane Sandy in 2012 "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." Such huge amounts of user-generated data contributed by disaster affected communities have become an important source of big crisis data for disaster response (Castillo, 2016; Reuter & Kaufhold, 2018), and at the same time have been used by the public at large to make sense of an event from social media (Stefan, Deborah, Milad, & Christian, 2018). Many research and practical studies have proved the value of social media data on disseminating warning and response information, enhancing situational awareness, facilitating allocation of resources, informing disaster risk reduction strategies and risk assessments (Watson, Finn, & Wadhwa, 2017; Reuter, Hughes, & Kaufhold, 2018; National Research Council, 2013), as well as fostering community resilience (Zhang, Drake, Li, Zobel, & Cowell, 2015). Despite these benefits, the challenges presented by the volume of the data still preclude large emergency organizations from using them routinely (Meier, 2013).

DOI: 10.4018/IJISCRAM.2018100101

Manually sifting through voluminous streaming data to filter useful information in real time is inherently impossible. Machine learning techniques show promising results in automating the process of identifying useful, relevant and trustworthy information in big crisis data (Qadir et al., 2016), despite many practical challenges (Mendoza, Poblete, & Castillo, 2010). Many works have successfully used supervised learning algorithms to automatically classify tweets (Caragea, Squicciarini, Stehle, Neppalli, & Tapia, 2014; Imran, Elbassuoni, Castillo, Diaz, & Meier, 2013). Supervised algorithms require labeled training data to learn classifiers that can be further used to label new data of the same type (also called test data). The labels generated for the test data are usually accurate when the training and the test data are drawn from the same distribution.

The requirements above result in two main challenges that machine learning algorithms face when used to classify user-generated tweets about emerging disasters such as floods, hurricanes, and terrorist attacks. First, labeled data is not easily available for an emergent "target" disaster for which a classifier is needed to help disaster response teams identify relevant tweets, and ultimately information useful for situational awareness. Labeling data is an expensive and time-consuming process, which does not provide a real-time solution for disaster response. Labeled data from a prior "source" disaster can potentially be used to learn a supervised classifier for the target disaster (Starbird, Palen, Hughes, & Vieweg, 2010). However, another challenge is posed by the fact that data from the "source" disaster and data from the target disaster may not share the same distribution (or characteristics), and the classifier learned from the source may not perform well on the target.

We attempt to address these problems by leveraging domain adaptation approaches that use both labeled source data and unlabeled target data, which is accumulating quickly during a disaster. More specifically, we define our problem as follows: given labeled tweets from a source domain, and unlabeled tweets from the target domain, the goal is to train a domain adaptation classifier to label tweets from the target domain. There are several types of approaches to address this problem (Pan & Yang, 2010). Most relevant to this work, there are approaches that use the unlabeled target data to change the representation or the distribution of the source data in a way that makes it possible to use classifiers learned from source to predict the target accurately (Daumé III, 2007; Sun, Feng, & Saenko, 2016). We refer to this type of approach as feature-based domain adaptation. Furthermore, there are approaches that learn classifiers from labeled source data and unlabeled target data using an Expectation Maximization (EM) type strategy (Dai, Xue, Yang, & Yu, 2007). Such approaches can be seen as parameter-based domain adaptation as they use parameters from the source domain to inform the parameter selection for the target domain. Correlation alignment algorithm (CORrelation ALignment, CORAL) (Sun et al., 2016), originally designed for images, is a feature-based adaptation approach that aligns the distribution of the source domain with the distribution of the target domain to reduce the variance shift. Self-training (Yarowsky, 1995) is a parameter-based adaptation approach. Previous works on using CORAL (Sopova, 2017) or self-training (Li, Caragea, Caragea, & Herndon, 2018) independently to identify disaster relevant tweets have shown promising results. In this paper, we have first compared CORAL with self-training to understand which approach benefits more from source adaptation based on unlabeled data. We have also designed a hybrid approach that combines CORAL with self-training, with the goal of improving the results obtained with each independent approach further. We have performed experimental evaluation of the task of identifying relevant tweets, which is the very first step in filtering disaster information. Our main contributions are as follows:

- We have compared a feature-based adaptation approach (specifically, CORAL) with a parameter-based adaptation approach (specifically, self-training), in the context of disaster tweet classification.
- We have proposed a hybrid feature-parameter based approach that combines CORAL and self-training adaptation approaches. The goal is to understand if the combined approach can improve the adaptation ability further by retaining the advantages of each individual approach.

- We have evaluated the combined feature-parameter based approach on pairs of source-target disasters from the CrisisLexT6 datasets (Olteanu, Castillo, Diaz, & Vieweg, 2014), and compared it with adaptation approaches that use the feature-based approach (CORAL) or the parameter-based adaptation (self-training) independently.

## RELATED WORK

Domain adaptation has attracted significant attention during the last decade. Domain adaptation has been studied both theoretically (Ben-David, Blitzer, Crammer, & Pereira, 2007; Blitzer, Crammer, Kulesza, Pereira, & Wortman, 2008) and in applications such as NLP tasks (Daumé III, 2007; Jiang & Zhai, 2007a), sentiment analysis (Blitzer, Dredze, & Pereira, 2007; Tan, Cheng, Wang, & Xu, 2009), text classification (Dai et al., 2007), Wifi location (Pan, Tsang, Kwok, & Yang, 2009), computer vision and object recognition (Sun, Feng, & Saenko, 2015), disaster tweets classification (Li et al., 2015; Li, Caragea, et al., 2018; Li et al., 2017; Mazloom, Li, Caragea, Imran, & Caragea, 2018). We review some simple but efficient and effective methods here, which can potentially be applied in real time, as applying domain adaptation on disaster-related data is time-critical.

Domain adaptation methods can be categorized into several classes based on different criteria. Methods in one class work by changing the representation of the source data to make it more similar to the representation of the target data (a.k.a., feature-based adaptation). For example, Daumé III (2007) proposed a very simple way to represent the source and target with duplicate features consisting of three versions: general version, source specific version and target specific version, under the assumption that some target labeled data $D_{Tl}$ is available, in addition to source labeled data $D_S$. A two-stage domain adaptation method was proposed by Jiang and Zhai (2007b): in the first stage a set of features generalizable across domains was identified, and in the second adaptation stage, a set of features specific to the target domain were selected. Structural correspondence learning (SCL) was proposed in (Blitzer et al., 2007), where the goal is to select pivot features based on common frequencies and also mutual information. Sun et al. (2015) also proposed a simple feature-adaptation method, the CORrelation ALignment (CORAL), to align the source with the target domain. CORAL performs the alignment by re-coloring whitened source features with the covariance of target distribution. Despite the simplicity of this method, the results were comparable with those of more complex, state-of-the-art algorithms, and this motivated us to use this approach for classifying disaster-related tweets.

Another class of methods is focused on classifiers that combine source labeled data with target unlabeled data, or even some small amounts of target labeled data, with different weights (a.k.a., instance-based adaptation). Jiang and Zhai (2007a) introduced an instance weighting framework for domain adaptation for use in NLP tasks. When the weighting is done at the parameter classifier level, domain adaptation approaches can be classified as parameter-based adaptation approaches. In this category, Dai et al. (2007) proposed a domain adaptation algorithm called Transfer Naïve Bayes classifier (NBTC) based on the Naïve Bayes algorithm and Expectation-maximization (EM), and used it to classify text documents. Tan et al. (2009) proposed a weighted version of the multinomial Naïve Bayes classifier combined with EM for sentiment analysis of text classification.

In recent years, several studies have focused on classifying disaster related/informative tweets with the goal of helping disaster response and recovery. Imran et al. (2016) researched the performance of the classifiers trained using different combinations of datasets obtained from past disasters. They performed extensive experimentation on real crisis datasets and showed that the past labels are useful when both source and target events are of the same type (e.g., two earthquakes). Li et al. (2015) proposed an unsupervised domain adaptation algorithm based on Expectation Maximization (EM) to classify disaster tweets. This approach made use of source labeled data together with target unlabeled data to build a weighted Naïve Bayes Classifier in an iterative way. The self-training procedure is run for a fixed number of iterations, or until convergence. Li, Caragea, et al. (2018) extended the EM-type approach with a self-training strategy and showed that self-training generally performs better. Li et al. (2017) also performed extensive experiments to provide practical guideline for the self-training domain adaptation. Mazloom et al. (2018) proposed a hybrid feature-instance adaptation approach, based on matrix factorization and the K-Nearest Neighbors algorithm. The proposed hybrid adaptation approach first applied matrix factorization to reduce the dimensionality of the data, and then used K-Nearest Neighbors algorithm to select a subset of the source disaster data that is representative for the target disaster. The selected subset was subsequently used to learn accurate Naïve Bayes classifiers for the target disaster. Experimental results showed that the approach could significantly improve the performance as compared with a baseline Naïve Bayes classifier.

As deep learning approaches become dominant in many NLP tasks, there are also studies that apply deep learning approaches to the area of disaster response. Caragea, Silvescu, and Tapia (2016) explored supervised Convolutional Neural Networks (CNN) to classify informative tweets from six flood events. Nguyen et al. (2016) used Convolutional Neural Networks (CNN) to classify crisis related tweets. They assumed that some target labeled data is available and used two simple supervised domain adaptation techniques to combine prior source disasters data with current disaster labeled data during training. One was weighting the prior source disasters data, while regularizing the modified model. The other was simply selecting a subset of the prior source disaster tweets, specifically those that were correctly labeled by a target-based classifier. Their experimental results showed that CNNs with simple instance selection domain adaptation technique gave better results. One drawback of these approaches is the requirement that some target labeled data is available. Li, Li, Caragea, and Caragea (2018) explored three types of word embeddings, as well as three sentence-encoding models to understand what embeddings/encodings are more suitable for use in crisis tweet classification tasks. The experimental results suggested that, for the traditional supervised learning setting, GloVe embeddings worked better than other approaches. Furthermore, GloVe embeddings trained on crisis data produce better results on more specific crisis tweet classification tasks (e.g., tweets informative versus non-informative), while GloVe embeddings pretrained on a large collection of general tweets produce better results on more general classification tasks (tweets relevant or not relevant to a crisis).

## CORRELATION ALIGNMENT ALGORITHM OVERVIEW

We adopt a simple, yet effective feature-based domain adaptation method, CORrelation ALignment (CORAL), introduced in Sun et al. (2016). CORAL works by aligning the distributions of the source and target data in an unsupervised manner. More specifically, CORAL minimizes the domain shift by aligning the second-order statistics of source and target distributions, namely, the covariance, without requiring any target labels. As stated in Sun et al. (2016), CORAL aligns the distributions by re-coloring whitened source features with the covariance of the target features. The approach involves the following steps:

1. compute covariance statistics in each domain
2. apply the whitening and re-coloring linear transformations to the source features.

After the source distribution is aligned to the target distribution, supervised learning proceeds as usual – a classifier is trained using the transformed source features and used to classify the target data. As the correlation alignment algorithm only changes the source features only, it can be used with any base classifier.

Formally, we are given source-domain training examples $D_S = \left\{\overrightarrow{x_i} \mid \overrightarrow{x_i} \in R^D, i = 1, \cdots, \left|S\right|\right\}$, with labels $L_S = \left\{y_i \mid i = 1, \cdots, \left|S\right|\right\}$, $y_i \in \left\{1, \cdots, L\right\}$, and target data $D_T = \left\{\overrightarrow{u_i} \mid \overrightarrow{u_i} \in R^D, i = 1, \cdots, \left|T\right|\right\}$. Here both $\left\{\vec{x}\right\}$ and $\left\{\vec{u}\right\}$ are $D$-dimensional feature representations $\varphi\left(I\right)$ of input instances $I$.

Suppose $\mu_S, \mu_T$ and $C_S, C_T$ are the feature vector means and covariance matrices for source $S$ and target $T$, respectively. According to Sun et al. (2016), to minimize the distance between the second-order statistics (covariance) of the source and target features, one can apply a linear transformation $A$ to the original source features. This transformation can be obtained as a solution to the following minimization problem:

$$\min_{A} C_{\hat{S}} - C_{T\,F}^{\,2} = \min_{A} A^T C_S A - C_{T\,F}^{\,2} \tag{1}$$

where $C_{\hat{S}}$ is the covariance of the transformed source features $D_S A$, and $\cdot_F^2$ denotes the Frobenius norm of a matrix, used as a distance metric. Let $D_S = U_S \pounds_S V_S^{\mathrm{T}}$ be the Singular Value Decomposition (SVD) of the matrix corresponding to the source data $D_S$. Similarly, let $D_T = U_T \pounds_T V_T^{\mathrm{T}}$ be the SVD decomposition of the matrix corresponding the target data $D_T$. Furthermore, let $\pounds_{T[1:r]}, U_{T[1:r]}, V_{T[1:r]}$ be the largest $r$ singular values and the corresponding left and right singular vectors of $D_T$, respectively. As proven in CORAL, the optimal solution for the above minimization problem can be found as:

$$A^* = \left( U_S \pounds_S^{+\frac{1}{2}} U_S^{\mathrm{T}} \right) \left( U_{T[1:r]} \pounds_{T[1:r]}^{+\frac{1}{2}} U_{T[1:r]}^{\mathrm{T}} \right) \tag{2}$$

which can be interpreted as follows: the first part is used to whiten the source data, while the second part is used to re-color it with the target covariance.

As Sun et al. (2016) suggest, after CORAL transforms the source features according to the target space, a classifier $f_{\vec{\omega}}$ parametrized by $\omega$ can be trained on the adjusted source features and directly applied to target features. In this work, we run experiments using Naïve Bayes Classifier. The CORAL algorithm is summarized in Algorithm 1.

### Algorithm 1: Correlation Alignment Algorithm

**Input**: Target unlabeled data $T$, source labeled data $S$

**Output**: Adjusted source labeled data $S^*$

import numpy as np
from scipy.linalg import fractional_matrix_power
$ncolsS = S.shape[1]$
$ncolsT = T.shape[1]$
$C_S = np.cov(S, rowvar = 0) + np.eye(ncolsS)$
$C_T = np.cov(T, rowvar = 0) + np.eye(ncolsT)$
$S = np.dot(S, fractional\_matrix\_power(C_S, -0.5))$
$S^* = np.dot(S, fractional\_matrix\_power(C_T, 0.5))$

## OVERVIEW OF THE NAÏVE BAYES DOMAIN ADAPTATION ALGORITHM WITH SELF-TRAINING

The Naïve Bayes domain adaptation with self-training builds a weighted Naïve Bayes (Manning, Raghavan, & Schütze, 2008) classifier, which linearly combines source and target data in an iterative fashion, to simultaneously estimate the prior $P\left(c_i\right)$ and the likelihood $P(w_j \mid c_i)$, as follows:

$$P\left(c_i\right) = \left(1-\gamma\right) P_S\left(c_i\right) + \gamma P_T\left(c_i\right) \quad (3)$$

$$P(w_j \mid c_i) = \left(1-\gamma\right) P_S(w_j \mid c_i) + \gamma P_T(w_j \mid c_i) \quad (4)$$

In the equations above, $c_i$ represents a class label, $w_j$ is a feature in the feature set or vocabulary $V$, the probability subscript ($S$ or $T$) denotes the type of data used to estimate that probability, i.e. $S$ denotes source labeled data, $T$ denotes target unlabeled data, and $\delta$ is a parameter that controls how fast we shift the weight from source to target data. This parameter is defined as $\gamma = \min(t* \delta,1)$, where $t = \left\{0,1,2,\cdots\right\}$ is the iteration number. Initially, $t = 0$, $\gamma = 0$, which means that only source labeled data is used. Then, according to the Bayes Theorem, we estimate the posterior class label $c_i$ for a new instance $d$ as:

$$P(c_i \mid d) \propto P\left(c_i\right) \prod_{j \in \left|V\right|} P(w_j \mid c_i) \quad (5)$$

At each iteration, the current classifier (originally trained only from source labeled data) is used to classify the remaining target unlabeled data (originally all the target unlabeled data). The most confidently classified unlabeled instances (e.g., top $k$ instances in each class) are moved to the training set, with hard (e.g., 0/1) labels, to be used in subsequent iterations. By default, the algorithm runs until convergence, where "convergence" means that the labels of the remaining target unlabeled instances don't change in between two consecutive iterations (Yarowsky, 1995). The domain adaptation approach with self-training is summarized in Algorithm 2.

The estimation of the priors $P\left(c_i\right)$ and likelihoods $P(w_j \mid c_i)$ is based on the Naïve Bayes classifier. Two variants of the Naïve Bayes classifier are used in the study. For binary bag-of-word representations (Mitchell, 1997), where the values of the features are discrete 0/1 values, Bernoulli Naive Bayes (Manning et al. 2008) is used. After the CORAL transformation, the values of the features become continuous, and the Gaussian Naive Bayes classifier (Mitchell, 2017) is used in that case. Specifically, Bernoulli Naïve Bayes algorithm estimates the class priors and likelihoods from the training data, using the add-1 smoothing strategy (to avoid zero probabilities), as follows:

*Algorithm 2:* **Naïve Bayes Domain Adaptation algorithm with self-training**

**Input**: Target unlabeled data $T$, source labeled data $S$, target test data $Tt$

**Output**: A Naïve Bayes classifier for target data and labels for instances in $T_{test}$

**Let** $T_{hard} = \phi$ and $T_{left} = T$, where $T_{hard}$ is the set of instances with hard labels assigned by self-training, and $T_{left}$ is the set of unlabeled instances left in $T$

**While** (labels assigned to instances in $T_{left}$ change) or (maximum number of iteration not reached) **do**

[**M-step:**] Simultaneously compute the prior and likelihood using Equations 3 and 4, respectively, using a combination of $S$ and $T_{hard}$ weighted based on $\gamma = \min\left(t^*\delta, 1\right)$, where $t$ is the iteration number; at the first iteration, only $S$ is used

[**E-step:**] Compute the posterior class probability of target instances still unlabeled, $T_{left}$, with Equation 5 and select the $k$ most confidently labeled instances from each class $c_i$ based on probability ranking, move them to $T_{hard}$ (to use for training in the next iteration) and remove them from $T_{left}$

3. Use the final classifier to predict the labels of the target test instances $T_{test}$

$$P\left(c_i\right) = \frac{N\left(c_i\right) + 1}{N + 1} \tag{6}$$

$$P(w_j = 0 \mid c_i) = \frac{N\left(w_j = 0, c_i\right) + 1}{N\left(c_i\right) + 2} \tag{7}$$

$$P(w_j = 1 \mid c_i) = \frac{N\left(w_j = 1, c_i\right) + 1}{N\left(c_i\right) + 2} \tag{8}$$

where $N$ is the total number of documents in the collection $D$, $N\left(c_i\right)$ is the number of documents in class $c_i$, $N\left(w_j = 0, c_i\right)$ is the number of documents in class $c_i$ that don't contain the word $w_j$, and $N\left(w_j = 1, c_i\right)$ is the number of documents in class $c_i$ that contain the word $w_i$.

For Gaussian Naïve Bayes, the estimation of priors is same as Equation 6, likelihood is assumed to Gaussian and is estimated with:

$$P(w_j \mid c_i) = \frac{1}{\sqrt{2\pi\delta_{ji}^2}} \mathrm{e}^{-\frac{(w_j - \mu_{ji})^2}{2\delta_{ji}^2}} \tag{9}$$

where $\mu_{ji}$ is the mean of feature $j$ of class $c_i$, $\delta_{ji}^2$ is the corresponding variance. Both of $\mu_{ji}$ and $\delta_{ji}^2$ are also estimated from the training set using maximum likelihood.

## FEATURE SELECTION

Sun et al. (2016) used a sentiment analysis dataset, where the dimensionality was reduced based on the information gain criterion. As we are interested in identifying informative target features and the information gain criterion requires labeled data, we use a different criterion for feature selection, specifically, an unsupervised feature selection algorithm called "Variance Threshold" (Scikit-Learn, 2016). Essentially, we remove all low-variance features from the target data. The low-variance features are defined as those that are either 0 or 1 in more than $k\,\%$ of the samples, which corresponds to the Variance Threshold equal to $0.k * \left(1 - 0.k\right)$. The Variance Threshold looks only at the features of $X$, but not at the class labels $y$. In order to select features, we first concatenate labeled source data

$S$ and unlabeled target data $T$ . Once a subset of the features is selected, we represent $S$ and $T$ using the selected features.

## DATA DESCRIPTION AND PREPROCESSING

We use the dataset CrisisLexT6 (Olteanu et al., 2014) in our experiments. The dataset consists of approximately 60,000 tweets posted during 6 crisis events in 2012 and 2013. The 60,000 tweets (about 10,000 in each disaster) have been labeled by crowdsourcing workers according to relatedness (as on-topic or off-topic). On-topic tweets are labeled as 1, and off-topic tweets as 0.

The tweets are preprocessed before they are used in training, domain adaptation and testing stages. The cleaning steps are the same as those used in (Li et al., 2015). For completeness, we summarize them in what follows:

- Non-printable, ASCII characters are removed, as they are 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 they are not informative for our classification tasks
- Duplicate tweets and empty tweets (that have no characters left after the cleaning) are removed

The numbers of tweets per class for each disaster, before and after cleaning, are presented in Table 1. After preprocessing, the source tweets are expressed via target features, i.e. via words that occur in the target tweets. The bag-of-words binary representation (Mitchell, 1997) is used to represent tweets as vectors of features.

## EXPERIMENTAL SETUP

We design our experiments to answer the following questions:

**Table 1. CrisisLeXT6 dataset**

| Data | | Before Cleaning | | | After Cleaning | | |
|---|---|---|---|---|---|---|---|
| **Abbreviation** | **Crisis** | **On-topic** | **Off-topic** | **Total** | **On-topic** | **Off-topic** | **Total** |
| SH | 2012 Sandy Hurricane | 6138 | 3870 | 10008 | 5261 | 3752 | 9013 |
| QF | 2013 Queensland Floods | 5414 | 4619 | 10033 | 3236 | 4550 | 7786 |
| BB | 2013 Boston Bombings | 5648 | 4364 | 10012 | 4441 | 4309 | 8750 |
| WT | 2013 West Texas Explosion | 5246 | 4760 | 10006 | 4123 | 4733 | 8856 |
| OT | 2013 Oklahoma Tornado | 4827 | 5165 | 9992 | 3209 | 5049 | 8258 |
| AF | 2013 Alberta Floods | 5189 | 4842 | 10031 | 3497 | 4714 | 8211 |

- How does the performance of the domain adaptation approaches compare with the performance of the supervised baselines?
- Do the domain adaptation approaches studied perform better with all features or with a reduced set of features?
- How does the feature-based adaptation (CORAL) perform compared to the parameter-based adaptation (self-training)?
- How does the hybrid feature-parameter adaptation approach compare with the individual feature-based and parameter-based approaches?

We perform the following experiments:

1. Run Bernoulli Naïve Bayes classifier on the source with the original features. This is the baseline for all experiments. We refer to this baseline as NB.
2. Run Bernoulli Naïve Bayes on the source represented using the features selected with VT. We refer to as NB+VT.
3. Run CORAL with the original features and transform the source labeled data according to the target unlabeled data. Use the transformed source to learn a Gaussian Naïve Bayes classifier. We refer to this experiment as NB+CORAL.
4. Run Variance Threshold (VT) on the combined dataset of $S$ and $T$ to select a subset of features. Run CORAL with the selected features to transform the source. Use the transformed source to learn a Gaussian Naïve Bayes classifier. We refer to this experiment as NB+CORAL+VT.
5. Run Bernoulli Naïve Bayes with self-training on the source with the original features. We refer to these experiments as NBST.
6. Run Variance Threshold (VT) on the combined dataset of $S$ and $T$ to select a subset of features. Run Bernoulli Naïve Bayes with self-training on the source represented using the features selected with VT. We refer to these experiments as NBST+VT.
7. Run Variance Threshold (VT) on the combined dataset of $S$ and $T$ to select a subset of features. Run CORAL with the selected features to transform the source. Run Gaussian Naïve Bayes with self-training on the transformed source. We refer to this experiment as NBST+CORAL+VT.

We setup the experiments as follows:

- Following the prior work in (Li et al., 2018a), we use 11 source-target pairs, as shown in Table 3. The pairs are formed by following the chronological order of the events in a pair (i.e., the source disaster happened before the target disaster).
- We perform 5-fold cross-validation over target and report the average accuracy over the 5 folds. With CORAL, each source is "aligned" with three target unlabeled folds, one target fold is used for testing, and one target fold is kept for future use as potential target labeled data. Similarly, NBST uses three target unlabeled folds in the training process, and one fold for testing.
- We varied the number of instances in the sources to see how the performance vary with different numbers of source instances. Concretely, in addition to the whole source labeled data, which we denote as Total, we also selected 500, 1000, 2000 instances from each class (on-topic or off-topic), respectively.
- The number of features selected varies from one experiment/split to another. For example, for one pair, the original dataset has 1334 features, and the VT approach selects anywhere from 160 to 176 features (for different splits), thus resulting in significant dimensionality reduction. In preliminary work, we varied the value of the threshold k in VT. Precisely, we experimented with k = 0.95, k = 0.90, k = 0.80. The highest accuracy was obtained when the threshold is equal to 0.99, and this is the value used in the experiments.

### Features Selected

To illustrate the features selected by the VT approach, Table 2 shows a subset of words/features that were kept after feature selection for 5 pairs, corresponding to the 5 target disasters used in our experiments. For each target disaster, the source disaster was chosen arbitrarily among the sources corresponding to that particular target. A source and target pair (S→T) is denoted using the source and target disaster abbreviations introduced in Table 1. All these five pairs use 1000 source instances (i.e., 500 instances per class). Table 2 shows selected features that are appear both in source and target tweets (second column), and also selected features that are specific to the target tweets (third column).

## RESULTS AND DISCUSSION

The evaluation of the classifiers is based on accuracy as the dataset is relatively balanced. The results of the experiments that are used to compare NB, with NB+VT, and with NB+CORAL and NB+CORAL+VT are presented in Table 3. The results of the experiments that are used to compare NBST, NBST+VT, and NB+CORAL+VT and NBST+CORAL+VT are shown in Table 4. In addition to accuracy, we also show the precision and recall scores for the NBST+CORAL+VT experiments in Table 5. In these tables, a source and target pair ( $S \rightarrow T$ ) is denoted using the source and target disaster abbreviations introduced in Table 1. The numbers 500, 1000, 2000 in the header denote how many source instances from each class (on-topic or off-topic) are used for training, and Total means that all source instances are used. The highlighted values are the best values for each pair with a certain number of source instances across different experiments/approaches. The more highlighted values one approach has, the better that approach performs. We also show the results using graphs for visual analysis. Specifically, Figure 1 shows the averaged accuracy over the 11 pairs used (with standard variation bars), and Figure 2 shows accuracy results obtained with the approaches considered for each pair separately. We use the results in these tables and figures to answer the research questions as follows.

How does the performance of the domain adaptation approaches compare with the performance of the supervised baselines?

Self-training with Naïve Bayes (NBST) has been shown to improve the performance of Naïve Bayes (NB) classifier in prior works ((Li, Caragea, et al., 2018; Li, Caragea, & Caragea, 2017). As can be seen in Table 4 and Figure 2, our results are consistent with what was found in prior works. Thus, our discussion regarding this question will focus more on the feature-based domain adaptation approach, i.e., CORAL. Specifically, in Table 3, we can compare the experimental results of NB+CORAL with the results of its NB baseline. Furthermore, we can perform a similar comparison when doing feature selection with VT, namely, the results of NB+CORAL+VT with the results of the baseline NB+VT. In general, it can be seen that applying CORAL contributes to an improvement in accuracy. NB+CORAL+VT gives the best results overall, regardless of the amount of source data used, although the NB baseline is better in a few cases. Naïve Bayes used with VT features only or with CORAL only does not perform well overall, as compared to the corresponding baselines.

Do the domain adaptation approaches studied perform better with all features or with a reduced set of features?

Specifically, we want to see whether CORAL and self-training perform better with the original features or with a reduced set of features obtained with VT. As can be seen in Table 3, when comparing the results of NB and NB+VT, overall, feature selection doesn't improve the results of the supervised Naïve Bayes used on the original features. Based on the results in Table 4, we investigate the usefulness of the VT feature selection in the context of NBST.

Specifically, we compare the results of NBST (which uses all the features) with the results of NBST+VT (which uses only the features selected based on VT). As can be seen, only for a small number of pairs, the results of NBST+VT are consistently better than the results of NBST, regardless

**Table 2. Examples of features selected by the VT approach**

| $S \rightarrow T$ | Features that appear both in source and target | Features that appear only in target |
|---|---|---|
| $SH \rightarrow QF$ | username, url, like, get, power, right, lol, people, new, everyone, one, go, know, love, see, time, stay, want, really, hope, good, safe, even, way, much, think, still, house, today, last, us, day, feel, home, via, affected, please, back, help, night, well, first, need, water,:(, thanks, oh, say, rain, haha, live,:), follow, news, fire, victims, two, south, river, floods, death, crisis, flood | queensland, bundaberg, #flood, brisbane, wales, #queensland, waters, #bigwet, australia, #brisbane, 'australia\'s', qld, #qldfloods, australian, braces, rises, #australia, toll, queensland's |
| $QF \rightarrow BB$ | username, url, new, get, like, people, victims, good, one, love, really, time, news, two, back, go, still, see, via, affected, last, please, day, think, right, live, would, us, everyone, help, need, want, even, home, heart, thoughts, world, shit, know, fuck, dead, today, says, old, city, prayers, man, video, never, lol, made, first, police, killed, died | #boston, boston, explosions, tragedy, bomb, suspect, marathon, bombings, #prayforboston, fbi, bombing, suspects, #bostonmarathon, explosion |
| $SH \rightarrow WT$ | username, url, sandy, like, get, right, lol, people, new, everyone, one, go, love, know, see, time, want, good, really, even, school, way, praying, would, still, bad, think, today, last, make, day, home, affected, back, via, please, god, need, help, first, prayers, say, man, never, live, pray, news, dead, victims, thoughts, many, fire, near, town, killed, video, another, texas, #west, plant, explosion, caught, west, massive | waco, explosion., #prayfortexas, bombing, tx, #westexplosion, #texas, reported, injured, texas., #westtx, fertilizer, boston, |
| $SH \rightarrow OT$ | username, hurricane, url, sandy, like, get, right, people, lol, new, everyone, one, go, love, know, see, time, stay, want, hit, going, hope, really, good, even, school, much, way, think, still, praying, today, last, take, make, day, home, feel, please, via, back, god, affected, come, help, first, need, thank, say, prayers, watch, city, relief, ok, never, victims, thoughts, lost, disaster, heart, dog | oklahoma,#prayforoklahoma,tornado,#oklahoma,oklahoma.,moore,#moore |
| $SH \rightarrow AF$ | username, url, like, get, right, power, people, lol, new, everyone, one, go, know, love, see, time, stay, want, good, really, hope, even, safe, much, way, still, better, think, today, last, make, take, day, home, affected, please, friends, back, well, need, night, help, first, come, water, thanks, high, say, work, thank, never, relief, city, great,:), two, many, email, red, river, flooding, best, floods, flood, #job, downtown | edmonton, calgary, canada, alberta, #yyc, #yycflood, #calgary, ab, #abflood |

of the number of source training instances used. Concretely, for 2 out of 11 pairs (SH → WT, BB → WT). It can also be observed that VT feature selection helps especially when smaller numbers of source labeled instances are used. When larger numbers of source labeled instances are available, NBST without feature selection performs better. Intuitively, larger datasets lead to better estimates for the likelihood probabilities of less frequent features, and, in effect, including those features results in better performance. However, the feature-based domain adaptation approach CORAL benefits from using feature selection. By comparing the results of NB+CORAL with the results of NB, we see that CORAL can improve the classification accuracy results, but not significantly. However, when we apply CORAL after doing feature selection, the accuracy improvements become much more significant, as can be seen from the comparison of the results of NB+VT with the results of NB+CORAL+VT. Thus, we can claim that performing VT feature selection, to remove low variance features, benefits CORAL. This may due to the fact that low variance features act as noise for both source and target, and therefore degrade the source whitening and recoloring effects.

How does the feature-based adaptation (CORAL) perform compared to the parameter-based adaptation (self-training)?

Based on results in Table 4 and Figure 2, we compare two domain adaptation methods, self-training and CORAL with Naïve Bayes as a base classifier, i.e., NB+CORAL+VT and NBST+VT. As can be seen, NBST+VT performs better than NB+CORAL+VT overall, and implicitly better than the baselines (NB, NB+VT). Furthermore, NBST is better than CORAL, as NBST benefits from using all features, while CORAL works better with a selected set of features.

How does the hybrid feature-parameter adaptation approach compare with the individual feature-based and parameter-based approaches?

To answer this question, we compare the results of NBST+CORAL+VT with the results of NBST and NBST+VT, and also with the results of NB+CORAL+VT in Table 4. As can be seen in Figures 1 and 2, overall the self-training approach performs better than the hybrid feature-parameter variant NBST+CORAL+VT. However, for some specific source-target pairs, the combined self-training and CORAL approach is better than either self-training or CORAL alone. More concretely, in Figure 2, we can see that for pairs SH → QF, QF → BB, SH → AF, the hybrid approach achieves the best performance. We can also see that NBST+CORAL+VT can improve the performance of NB+CORAL+VT for almost all pairs, regardless of the number of source instances used. As last, from Table 5, we can see that NBST+CORAL+VT performs well also in terms of precision and recall metrics, with an average precision of 0.836 and average recall of 0.827 when all source instances are used for training.

## CONCLUSION

Domain adaptation has become crucial for many machine learning applications, as it enables the use of unlabeled data in domains where labeled data is not available. This is especially true in the case of social media analysis in the context of emerging disasters. Therefore, in this paper, we compared different domain adaptation approaches that make use of labeled tweets from a prior source disaster and unlabeled tweets from the target disaster to train a classifier for the target disaster. We used 11 source and target disaster pairs to evaluate these approaches.

We first compared a feature-based adaptation approach, CORAL, with a parameter-based adaptation approach, self-training, in the context of disaster-related tweet classification. Naïve Bayes was used as a base classifier for both adaptation approaches. CORAL is a simple yet effective domain adaptation method based on unsupervised feature alignment between source and target data. Naïve Bayes self-training (NBST) builds a weighted Naïve Bayes classifier iteratively by combining source labeled data and target unlabeled data. To understand if the two approaches have complementary strengths that can be combined, we also designed a hybrid feature-parameter adaptation approach, which combines CORAL with self-training, and compared it with the individual CORAL and self-training adaptation approaches. Experimental results showed that the domain shift with CORAL, from source data to target data, generally improves the performance of the classifiers trained on source. More specifically, CORAL combined with VT feature selection results in classifiers that have higher performance when compared with the classifiers learned from the original source data. The comparison between CORAL and Naïve Bayes self-training showed that the later approach performs better than CORAL in many cases, especially when leveraging more features. It can be hypothesized that the gradual labeling and usage of the target data in NBST can potentially capture more knowledge from the target unlabeled data, as compared to the one-shot use of the target unlabeled data in CORAL (to shift the distribution of the source data). The comparison of the hybrid feature-parameter adaptation approach with the individual feature-based and parameter-based adaptation approaches supports this hypothesis. Specifically, when comparing the hybrid approach with the individual feature-based and parameter-based approaches, no significant gains are observed from the combination. Furthermore, the results of the hybrid are overall similar with the results of the NBST approach itself. This suggests that the CORAL approach, while useful on its own, does not have complementary strengths as compared with NBST, which is a better approach overall. Nevertheless, there are some specific pairs where the hybrid approach performs better than the self-training approach.

As part of future work, it is of interest to extend the proposed feature-parameter adaptation into a feature-instance-parameter adaptation, by including an additional instance selection step as in (Mazloom et al., 2018). It is also of interesting to explore deep domain adaptation approaches on disaster tweet classification tasks, for example, marginalized Stacked Denoising Auto-encoders (mSDA) (Chen, Xu, Weinberger, & Sha, 2012) and domain adversarial neural networks (Ajakan, Germain, Larochelle, Laviolette, & Marchand, 2014; Ganin et al., 2015), and other similar models proposed for NLP tasks or computer vision tasks. Furthermore, we plan to explore an approach closely related to CORAL, specifically Deep CORAL (Sun & Saenko, 2016), which extends CORAL to learn a nonlinear transformation that aligns correlations of layer activations in deep neural networks.

**Table 3. Accuracy results of CORAL with the original features (NB+CORAL) and CORAL with the Variance Threshold (VT) features selection (NB+CORAL+VT), together with accuracy results for baselines Naïve Bayes (NB) and Naïve Bayes with VT features selection (NB+VT).**

| | $S \rightarrow T$ | 500 | 1000 | 2000 | Total | | $S \rightarrow T$ | 500 | 1000 | 2000 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NB | $SH \rightarrow QF$ | 0.692 | 0.787 | 0.810 | 0.772 | NB + CORAL | $SH \rightarrow QF$ | 0.695 | 0.749 | 0.800 | 0.827 |
| | $SH \rightarrow BB$ | 0.696 | 0.763 | 0.723 | 0.695 | | $SH \rightarrow BB$ | 0.692 | 0.742 | 0.769 | **0.780** |
| | $QF \rightarrow BB$ | 0.714 | 0.728 | 0.731 | **0.747** | | $QF \rightarrow BB$ | 0.614 | 0.611 | 0.603 | 0.571 |
| | $SH \rightarrow WT$ | 0.679 | 0.754 | 0.755 | 0.774 | | $SH \rightarrow WT$ | 0.702 | **0.767** | **0.800** | **0.848** |
| | $BB \rightarrow WT$ | 0.922 | 0.931 | 0.934 | 0.948 | | $BB \rightarrow WT$ | 0.770 | 0.816 | 0.869 | 0.908 |
| | $SH \rightarrow OT$ | 0.773 | 0.805 | 0.836 | **0.815** | | $SH \rightarrow OT$ | 0.696 | 0.756 | 0.807 | 0.812 |
| | $QF \rightarrow OT$ | 0.811 | 0.821 | 0.833 | **0.838** | | $QF \rightarrow OT$ | 0.630 | 0.639 | 0.647 | 0.628 |
| | $BB \rightarrow OT$ | **0.819** | 0.813 | **0.834** | **0.846** | | $BB \rightarrow OT$ | 0.684 | 0.627 | 0.700 | 0.801 |
| | $SH \rightarrow AF$ | 0.699 | 0.737 | 0.748 | 0.714 | | $SH \rightarrow AF$ | 0.670 | 0.733 | 0.772 | 0.796 |
| | $QF \rightarrow AF$ | **0.759** | 0.764 | 0.781 | 0.788 | | $QF \rightarrow AF$ | 0.666 | 0.645 | 0.684 | 0.657 |
| | $BB \rightarrow AF$ | 0.710 | **0.716** | 0.734 | 0.742 | | $BB \rightarrow AF$ | 0.643 | 0.573 | 0.599 | 0.650 |
| | **Average** | 0.752 | 0.784 | 0.793 | 0.789 | | **Average** | 0.678 | 0.696 | 0.732 | 0.753 |
| NB+ VT | $S \rightarrow T$ | 500 | 1000 | 2000 | Total | NB + CORAL + VT | $S \rightarrow T$ | 500 | 1000 | 2000 | Total |
| | $SH \rightarrow QF$ | 0.645 | 0.763 | 0.773 | 0.724 | | $SH \rightarrow QF$ | **0.751** | **0.851** | **0.852** | **0.838** |
| | $SH \rightarrow BB$ | 0.694 | **0.766** | 0.703 | 0.686 | | $SH \rightarrow BB$ | **0.789** | 0.764 | **0.827** | 0.766 |
| | $QF \rightarrow BB$ | 0.712 | 0.703 | 0.716 | 0.717 | | $QF \rightarrow BB$ | **0.834** | **0.819** | **0.804** | 0.680 |
| | $SH \rightarrow WT$ | 0.682 | 0.735 | 0.714 | 0.738 | | $SH \rightarrow WT$ | **0.802** | 0.738 | 0.674 | 0.836 |
| | $BB \rightarrow WT$ | **0.923** | 0.931 | 0.931 | 0.942 | | $BB \rightarrow WT$ | 0.888 | **0.945** | **0.945** | **0.949** |
| | $SH \rightarrow OT$ | 0.763 | 0.768 | 0.795 | 0.762 | | $SH \rightarrow OT$ | **0.853** | **0.858** | **0.857** | 0.753 |
| | $QF \rightarrow OT$ | 0.796 | 0.801 | 0.815 | 0.815 | | $QF \rightarrow OT$ | **0.827** | **0.867** | **0.873** | 0.815 |
| | $BB \rightarrow OT$ | 0.796 | 0.791 | 0.806 | 0.808 | | $BB \rightarrow OT$ | 0.792 | **0.853** | 0.827 | 0.823 |
| | $SH \rightarrow AF$ | 0.669 | 0.701 | 0.715 | 0.651 | | $SH \rightarrow AF$ | **0.770** | **0.843** | **0.857** | **0.846** |
| | $QF \rightarrow AF$ | 0.740 | 0.740 | 0.742 | 0.748 | | $QF \rightarrow AF$ | 0.732 | **0.807** | **0.813** | **0.802** |
| | $BB \rightarrow AF$ | 0.685 | 0.694 | 0.697 | 0.695 | | $BB \rightarrow AF$ | **0.712** | 0.713 | **0.744** | **0.791** |
| | **Average** | 0.737 | 0.763 | 0.764 | 0.753 | | **Average** | **0.795** | **0.823** | **0.825** | **0.809** |

**Table 4. Accuracy results of domain adaptation approaches, Naïve Bayes with self-training (NBST), and Naïve Bayes with self-training and Variance Threshold (VT) feature selection (NBST+VT), CORAL with VT feature selection (NB+CORAL+VT), and the hybrid feature-parameter based approach (self-training on top of CORAL) with VT feature selection (NBST+CORAL+VT).**

| | $S \rightarrow T$ | 500 | 1000 | 2000 | Total | | $S \rightarrow T$ | 500 | 1000 | 2000 | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NB ST | $SH \rightarrow QF$ | 0.824 | 0.843 | 0.855 | 0.824 | NB + CORAL + VT | $SH \rightarrow QF$ | 0.751 | 0.851 | 0.852 | 0.838 |
| | $SH \rightarrow BB$ | **0.828** | **0.832** | **0.846** | **0.841** | | $SH \rightarrow BB$ | 0.789 | 0.764 | 0.827 | 0.766 |
| | $QF \rightarrow BB$ | 0.787 | 0.792 | 0.814 | **0.819** | | $QF \rightarrow BB$ | 0.834 | 0.819 | 0.804 | 0.680 |
| | $SH \rightarrow WT$ | 0.923 | 0.924 | 0.923 | 0.908 | | $SH \rightarrow WT$ | 0.802 | 0.738 | 0.674 | 0.836 |
| | $BB \rightarrow WT$ | 0.929 | 0.928 | 0.939 | 0.948 | | $BB \rightarrow WT$ | 0.888 | 0.945 | 0.945 | 0.949 |
| | $SH \rightarrow OT$ | 0.858 | **0.879** | **0.887** | **0.878** | | $SH \rightarrow OT$ | 0.853 | 0.858 | 0.857 | 0.753 |
| | $QF \rightarrow OT$ | **0.853** | 0.850 | 0.855 | **0.855** | | $QF \rightarrow OT$ | 0.827 | 0.867 | 0.873 | 0.815 |
| | $BB \rightarrow OT$ | **0.848** | **0.848** | **0.865** | **0.869** | | $BB \rightarrow OT$ | 0.792 | 0.853 | 0.827 | 0.823 |
| | $SH \rightarrow AF$ | 0.793 | 0.799 | 0.840 | 0.826 | | $SH \rightarrow AF$ | 0.770 | 0.843 | 0.857 | 0.846 |
| | $QF \rightarrow AF$ | 0.824 | 0.830 | **0.852** | **0.860** | | $QF \rightarrow AF$ | 0.732 | 0.807 | 0.813 | 0.802 |
| | $BB \rightarrow AF$ | 0.802 | **0.824** | **0.839** | **0.840** | | $BB \rightarrow AF$ | 0.712 | 0.713 | 0.744 | 0.791 |
| | **Average** | 0.843 | 0.850 | **0.865** | **0.861** | | **Average** | 0.795 | 0.823 | 0.825 | 0.809 |
| NB ST+ VT | $S \rightarrow T$ | 500 | 1000 | 2000 | Total | NBST + CORAL + VT | $S \rightarrow T$ | 500 | 1000 | 2000 | Total |
| | $SH \rightarrow QF$ | **0.844** | 0.845 | 0.839 | 0.806 | | $SH \rightarrow QF$ | 0.815 | **0.891** | **0.880** | **0.877** |
| | $SH \rightarrow BB$ | 0.827 | 0.830 | 0.845 | 0.815 | | $SH \rightarrow BB$ | 0.756 | 0.765 | 0.824 | 0.763 |
| | $QF \rightarrow BB$ | 0.798 | 0.805 | 0.812 | **0.819** | | $QF \rightarrow BB$ | **0.845** | **0.859** | **0.848** | 0.743 |
| | $SH \rightarrow WT$ | **0.936** | **0.936** | **0.928** | **0.923** | | $SH \rightarrow WT$ | 0.805 | 0.881 | 0.791 | 0.906 |
| | $BB \rightarrow WT$ | **0.944** | **0.946** | **0.950** | **0.953** | | $BB \rightarrow WT$ | 0.875 | 0.921 | 0.945 | 0.936 |
| | $SH \rightarrow OT$ | **0.868** | 0.874 | 0.875 | 0.866 | | $SH \rightarrow OT$ | 0.863 | 0.878 | 0.880 | 0.735 |
| | $QF \rightarrow OT$ | 0.832 | 0.835 | 0.843 | 0.842 | | $QF \rightarrow OT$ | 0.821 | **0.887** | **0.878** | 0.815 |
| | $BB \rightarrow OT$ | 0.845 | 0.840 | 0.844 | 0.843 | | $BB \rightarrow OT$ | 0.828 | 0.823 | 0.854 | 0.821 |
| | $SH \rightarrow AF$ | 0.815 | 0.818 | 0.818 | 0.797 | | $SH \rightarrow AF$ | **0.849** | **0.876** | **0.879** | **0.861** |
| | $QF \rightarrow AF$ | **0.827** | **0.832** | 0.846 | 0.851 | | $QF \rightarrow AF$ | 0.751 | 0.824 | 0.830 | 0.835 |
| | $BB \rightarrow AF$ | **0.805** | 0.805 | 0.817 | 0.822 | | $BB \rightarrow AF$ | 0.747 | 0.797 | 0.757 | 0.811 |
| | **Average** | **0.849** | 0.851 | 0.856 | 0.849 | | **Average** | 0.814 | **0.855** | 0.851 | 0.828 |

**Table 5. Precision and recall scores of the experiments with the hybrid feature-parameter based approach (self-training on top of CORAL) with VT feature selection (NBST+CORAL+VT)**

| | $S \rightarrow T$ | Precision | | | | Recall | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 500 | 1000 | 2000 | Total | 500 | 1000 | 2000 | Total |
| NBST + CORAL + VT | $SH \rightarrow QF$ | 0.828 | 0.896 | 0.886 | 0.880 | 0.816 | 0.890 | 0.880 | 0.876 |
| | $SH \rightarrow BB$ | 0.772 | 0.774 | 0.828 | 0.776 | 0.754 | 0.764 | 0.824 | 0.762 |
| | $QF \rightarrow BB$ | 0.846 | 0.862 | 0.852 | 0.752 | 0.846 | 0.858 | 0.848 | 0.742 |
| | $SH \rightarrow WT$ | 0.820 | 0.884 | 0.790 | 0.910 | 0.806 | 0.880 | 0.792 | 0.906 |
| | $BB \rightarrow WT$ | 0.878 | 0.924 | 0.944 | 0.936 | 0.876 | 0.922 | 0.944 | 0.936 |
| | $SH \rightarrow OT$ | 0.870 | 0.884 | 0.882 | 0.748 | 0.862 | 0.880 | 0.880 | 0.734 |
| | $QF \rightarrow OT$ | 0.836 | 0.888 | 0.880 | 0.834 | 0.820 | 0.886 | 0.878 | 0.816 |
| | $BB \rightarrow OT$ | 0.832 | 0.848 | 0.858 | 0.828 | 0.826 | 0.822 | 0.854 | 0.820 |
| | $SH \rightarrow AF$ | 0.860 | 0.882 | 0.892 | 0.876 | 0.848 | 0.876 | 0.878 | 0.862 |
| | $QF \rightarrow AF$ | 0.782 | 0.826 | 0.838 | 0.840 | 0.754 | 0.824 | 0.828 | 0.834 |
| | $BB \rightarrow AF$ | 0.748 | 0.842 | 0.760 | 0.814 | 0.748 | 0.796 | 0.758 | 0.810 |
| | Average | 0.825 | 0.865 | 0.855 | 0.836 | 0.814 | 0.854 | 0.851 | 0.827 |

**Figure 1. Average accuracy results (with standard derivation) over all pairs, for all approaches studied, across different numbers of source training instances**

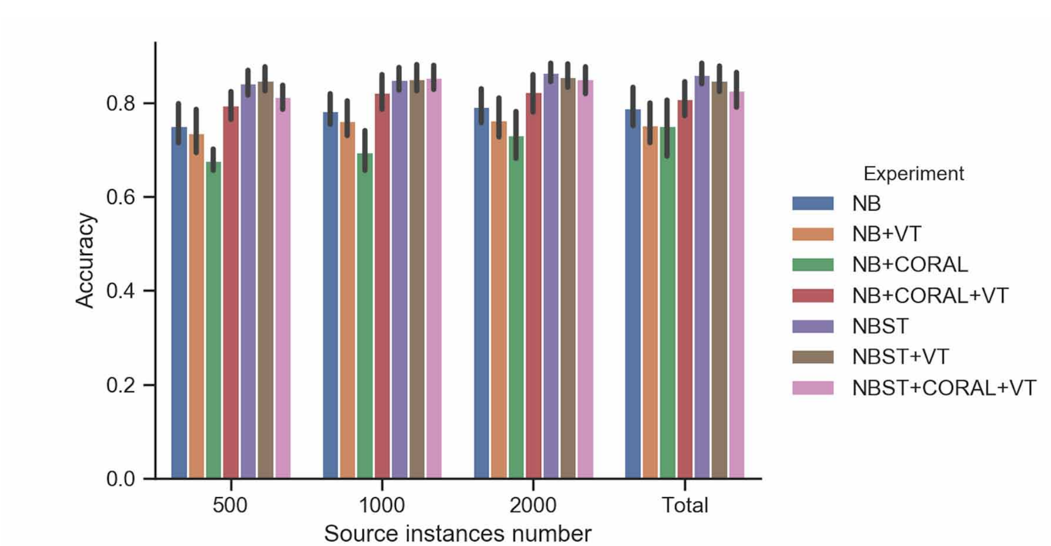
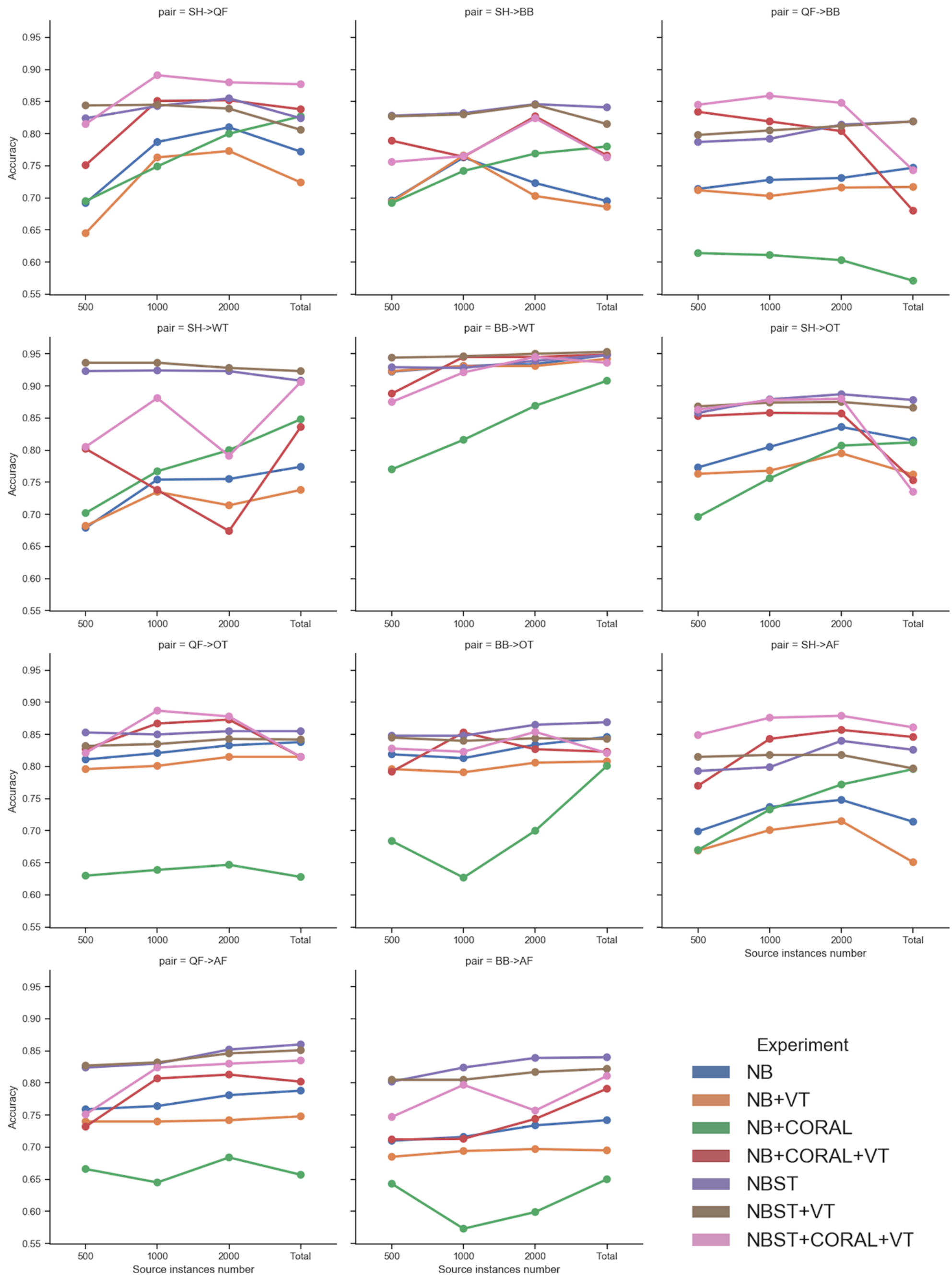


**Figure 2. Accuracy results for all 11 pairs and all approaches compared, across different amounts of source training data**



## REFERENCES

Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., & Marchand, M. (2014). Domain-adversarial neural networks. Retrieved from http://arxiv.org/abs/1412.4446

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In Advances in neural information processing systems. Cambridge, MA: MIT Press.

Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Wortman, J. (2008). Learning bounds for domain adaptation. In Advances in neural information processing systems. Cambridge, MA: MIT Press.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. Association for computational linguistics.

Caragea, C., Silvescu, A., & Tapia, A. H. (2016). Identifying informative messages in disasters using convolutional neural networks. In *Proceedings of the 13th proceedings of the international conference on information systems for crisis response and management*, Rio de Janeiro, Brasil, May 22-25. Retrieved from http://idl.iscram.org/files/corneliacaragea/2016/1397C orneliaCarageaetal2016.pdf

Caragea, C., Squicciarini, A. C., Stehle, S., Neppalli, K., & Tapia, A. H. (2014). Mapping moods: Geo-mapped sentiment analysis during hurricane sandy. In *Proceedings of the 11th proceedings of the international conference on information systems for crisis response and management*, University Park, PA, May 18-21.

Castillo, C. (2016). *Big crisis data: Social media in disasters and time-critical situations*. Cambridge University Press. doi:10.1017/CBO9781316476840

Chen, M., Xu, Z. E., Weinberger, K. Q., & Sha, F. (2012). Marginalized denoising autoencoders for domain adaptation. Retrieved from http://arxiv.org/abs/1206.4683

Dai, W., Xue, G.-R., Yang, Q., & Yu, Y. (2007). Transferring naïve bayes classifiers for text classification. In *Proceedings of the 22nd national conference on artificial intelligence* (Vol. 1, pp. 540–545). AAAI Press.

Daumé, H. III. (2007, June). Frustratingly easy domain adaptation. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 256–263). Prague, Czech Republic: Association for Computational Linguistics.

Meier, P. (2013, May 2). Crisis maps: Harnessing the power of big data to deliver humanitarian assistance. *Forbes*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., . . . Lempitsky, V. S. (2015). Domain-adversarial training of neural networks. Retrieved from http://arxiv.org/abs/1505.07818

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013). Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd international world wide web conference, WWW '13*, Rio de Janeiro, Brazil, May 13-17 (pp. 1021–1024). doi:10.1145/2487788.2488109

Imran, M., Mitra, P., & Srivastava, J. (2016). Cross-language domain adaptation for classifying crisis-related short messages. In Proceedings of the ISCRAM 2016 conference, Rio de Janeiro, Brazil. Academic Press.

Jiang, J., & Zhai, C. (2007a, June). Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 264–271). Prague, Czech Republic: Association for Computational Linguistics.

Jiang, J., & Zhai, C. (2007b). A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the sixteenth ACM conference on conference on information and knowledge management* (pp. 401–410). ACM. doi:10.1145/1321440.1321498

Li, H., Caragea, D., & Caragea, C. (2017). Towards practical usage of a domain adaptation algorithm in the early hours of a disaster. In *Proceedings of the 14th ISCRAM conference, Albi, France*, May. Academic Press.

Li, H., Caragea, D., Caragea, C., & Herndon, N. (2018). Disaster response aided by tweet classification with a domain adaptation approach. *Journal of Contingencies and Crisis Management*, *26*(1), 16–27. doi:10.1111/1468-5973.12194

Li, H., Guevara, N., Herndon, N., Caragea, D., Neppalli, K., Caragea, C., . . . Tapia, A. H. (2015). Twitter Mining for Disaster Response: A Domain Adaptation Approach. In *Proceedings of the 12th international conference on information systems for crisis response and management.* Academic Press.

Li, H., Li, X., Caragea, D., & Caragea, C. (2018). Comparison of word embeddings and sentence encodings as generalized representations for crisis tweet classification tasks. Retrieved from https://www.cs.uic.edu/cornelia/papers/iscramasian18.pdf

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. New York, NY: Cambridge University Press. doi:10.1017/CBO9780511809071

Maron, D. F. (2013). How social media is changing disaster response. *Scientific American*. Retrieved from https://www.scientificamerican.com/article/how-social-media-is-changing-disaste

Mazloom, R., Li, H., Caragea, D., Imran, M., & Caragea, C. (2018). Classification of twitter disaster data using a hybrid feature-instance adaptation approach. Retrieved from http://mimran.me/papers/MazloometalISCRAM 2018.pdf

Mendoza, M., Poblete, B., & Castillo, C. (2010). Twitter under crisis: Can we trust what we rt? In *Proceedings of the first workshop on social media analytics (soma '10)* (pp. 71–79). Academic Press.

Mitchell, T. (1997). *Machine learning*. McGraw-Hill Science/Engineering/Math.

Mitchell, T. (2017). Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression. Retrieved from http://www.cs.cmu.edu/ tom/mlbook/NBayesLogReg.pdf

National Research Council. (2013). *Public response to alerts and warnings using social media: Report of a workshop on current knowledge and research gaps.* Washington, DC: The National Academies Press. Retrieved from https://www.nap.edu/catalog/15853/public-response-to-alerts-and-warnings-using-

Nguyen, D. T., Al-Mannai, K., Joty, S. R., Sajjad, H., Imran, M., & Mitra, P. (2016). Rapid classification of crisis-related data on social networks using convolutional neural networks. Retrieved from http://arxiv.org/abs/1608.03902

Olteanu, A., Castillo, C., Diaz, F., & Vieweg, S. (2014). Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proceedings of the eighth international conference on weblogs and social media, ICWSM 2014,* Ann Arbor, Michigan, June 1-4. Academic Press. Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8091

Pan, S. J., Tsang, I. W., Kwok, J. T., & Yang, Q. (2009). Domain adaptation via transfer component analysis. In *Proceedings of the 21st international joint conference on artificial intelligence* (pp. 1187–1192). Academic Press.

Pan, S. J., & Yang, Q. (2010, October). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. doi:10.1109/TKDE.2009.191

Qadir, J., Ali, A., Rasool, R. U., Zwitter, A., Sathiaseelan, A., & Crowcroft, J. (2016). Crisis analytics: Big data driven crisis response.

Reuter, C., Hughes, A. L., & Kaufhold, M.-A. (2018). Social media in crisis management: An evaluation and analysis of crisis informatics research. *International Journal of Human-Computer Interaction*, *34*(4), 280–294. doi:10.1080/10447318.2018.1427832

Reuter, C., & Kaufhold, M.-A. (2018). Fifteen years of social media in emergencies: A retrospective review and future directions for crisis informatics. *Journal of Contingencies and Crisis Management*, *26*(1), 41–57. doi:10.1111/1468-5973.12196

Scikit-Learn. (2016). Variance threshold. Retrieved from http://scikit-learn.org

Sopova, O. (2017). *Domain adaptation for classifying disaster-related twitter data*. Manhattan, KS: Kansas State University.

Starbird, K., Palen, L., Hughes, A., & Vieweg, S. (2010). Chatter on the red: What hazards threat reveals about the social life of microblogged information. In *Proceedings of the ACM 2008 conference on computer supported cooperative work (CSCW2010)* (pp. 241–250). Academic Press.

Stefan, S., Deborah, B., Milad, M., & Christian, E. (2018). Sense-making in social media during extreme events. *Journal of Contingencies and Crisis Management*, *26*(1), 4–15. doi:10.1111/1468-5973.12193

Sun, B., Feng, J., & Saenko, K. (2016). Return of frustratingly easy domain adaptation. In *Proceedings of the thirtieth AAAI conference on artificial intelligence (AAAI-16)* (pp. 2058–2065). AAAI.

Sun, B., & Saenko, K. (2016). Deep CORAL: correlation alignment for deep domain adaptation. Retrieved from http://arxiv.org/abs/1607.01719

Tan, S., Cheng, X., Wang, Y., & Xu, H. (2009). Adapting Naïve Bayes to domain adaptation for sentiment analysis. In *Proceedings of the 31th European conference on IR research on advances in information retrieval* (pp. 337–349). Springer-Verlag. doi:10.1007/978-3-642-00958-7_31

Watson, H., Finn, R. L., & Wadhwa, K. (2017). Organizational and societal impacts of big data in crisis management. *Journal of Contingencies and Crisis Management*, *25*(1), 15–22. doi:10.1111/1468-5973.12141

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on association for computational linguistics* (pp. 189-196). Stroudsburg, PA: Association for Computational Linguistics. doi:10.3115/981658.981684

Zhang, Y., Drake, W., Li, Y., Zobel, C. W., & Cowell, M. (2015). Fostering community resilience through adaptive learning in a social media age: Municipal twitter use in new jersey following hurricane sandy. In L. Palen, M. Büscher, T. Comes, & A. L. Hughes (Eds.), *12th proceedings of the international conference on information systems for crisis response and management*, Krystiansand, Norway, May 24-27. ISCRAM Association.

*Hongmin Li, PhD candidate, Department of Computer Science at Kansas State University. Her research interests are mainly related to machine learning more specifically domain adaptation approaches on big crisis social media data.*

*Oleksandra Sopova is a software engineer at Microsoft, applying machine learning for Microsoft Search in Office 365. Her interests include natural language processing, deep learning, and anomaly detection. During grad school, Oleksandra has published research on Domain Adaptation in Machine Learning under the Fulbright Scholarship.*

*Doina Caragea, Ph.D., is a Professor at Kansas State University. Her research and teaching interests are in the areas of machine learning, data mining, data science, information retrieval and text mining, with applications to crisis informatics, security informatics, recommender systems, and bioinformatics. Her projects build upon close collaborations with social scientists, security experts and life scientists, and aim to provide practical computational approaches to address real-world challenges. Dr. D. Caragea received her PhD in Computer Science from Iowa State University in August 2004, and was honored with the Iowa State University Research Excellence Award for her work. She has published more than 100 refereed conference and journal articles. Her research has been supported by several NSF grants.*

*Cornelia Caragea is an Associate Professor of Department of Computer Science, University of Illinois at Chicago, and also an Adjunct Associate Professor of Department of Computer Science, Kansas State University. Her research interests are in artificial intelligence, machine learning, information retrieval, and natural language processing.*