Classification of Twitter Disaster Data Using a Hybrid Feature-Instance Adaptation Approach

by

Reza Mazloom

B.S., Staffordshire University, 2013

A Thesis

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

2018

Approved by:

Major Professor
Doina Caragea

Copyright

Abstract

Huge amounts of data that are generated on social media during emergency situations are regarded as troves of critical information. The use of supervised machine learning techniques in the early stages of a disaster is challenged by the lack of labeled data for that particular disaster. Furthermore, supervised models trained on labeled data from a prior disaster may not produce accurate results. To address these challenges, domain adaptation approaches, which learn models for predicting the target, by using unlabeled data from the target disaster in addition to labeled data from prior source disasters, can be used. However, the resulting models can still be affected by the variance between the target domain and the source domain. In this context, we propose to use a hybrid feature-instance adaptation approach based on matrix factorization and the k-nearest neighbors algorithm, respectively. The proposed hybrid adaptation approach is used to select a subset of the source disaster data that is representative of the target disaster. The selected subset is subsequently used to learn accurate supervised or domain adaptation Naïve Bayes classifiers for the target disaster. In other words, this study focuses on transforming the existing source data to bring it closer to the target data, thus overcoming the domain variance which may prevent effective transfer of information from source to target. A combination of selective and transformative methods are used on instances and features, respectively. We show experimentally that the proposed approaches are effective in transferring information from source to target. Furthermore, we provide insights with respect to what types and combinations of selections/transformations result in more accurate models for the target.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I like to thank my thesis advisor Dr. Doina Caragea of the Computer Science Department at Kansas State University. She has been a great mentor and I have received great knowledge, skills and life lessons from her, allowing me to grow as a individual while working towards my Master's Degree. I am truly grateful for her kindness and patience when guiding my pursuit of higher education. Her approach will be one of my guides when I become a professor or parent myself. I would like to thank my fellow researchers who have contributed to the Machine Learning Twitter project over the years especially HongMin Li, whose work has guided me along the way.

I would also like to thank my committee members Dr. Daniel Andresen and Dr. Torben Amtoft of the Computer Science Department at Kansas State University for their insight and guidance. I have learned distributed computing and algorithms from these professors. These algorithms already have influenced and will continue to influence my research career. I'm also grateful for their valuable comments and guidance they have given me during my degree.

I would also like to thank Dr. Sue Brown of the Biology Division at Kansas State University for assisting in funding my Degree. I will never forget the caring attitude Dr. Brown puts into the development of her students and her work. Diving into an interdisciplinary field and working with the Bioinformatics Center staff at Kansas State University has given me many biological points of view and taught me how communication is the cornerstone of any interdisciplinary and interdepartmental project. This Degree was supported by an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under grant number P20 GM103418.

I would also like to thank Dr. Majid Jaberi-Douraki of the Mathematics and Anatomy and Physiology Department of Kansas State University for giving me the opportunity to engage in professional applied research. The experience with the 1Data Project and Jaberi

Group has provided me with experiences that will greatly influence my future decisions after finishing my studies. I have been grateful for working on the 1Data project with so much potential meeting the developers, Dr. Jim Riviere from the College of Veterinary Medicine and Dr. Gerald J. Wyckoff at UMKC. I am looking forward to the impact 1Data will have on animal and human cross-species health analysis. This degree was assisted by KCALSI and Hall Family Foundation Award for Strategic Grant Program in KC and Elanco Foundation.

I would like to thank Kansas State University's High Performance Computing team, specifically Dr. Dave Turner and Adam Tygart. The computing for this project was performed on the Beocat Research Cluster at Kansas State University. I thank the National Science Foundation for support from grant CNS-1429316, which partially funded the cluster. I also thank the National Science Foundation for support from the grants IIS-1802284, IIS-1741345, IIS-1526542 and CMMI-1541155. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either express or implied, of the National Science Foundation.

Finally, I would like to express my profound gratitude to my parents, family and friends for supporting and encouraging me throughout the years of my study and research. This accomplishment would not have been possible without their continuous support. Thank you.

Author

Reza Mazloom

Dedication

To my father, Dr. Hamid Mazloom, and my mother, Dr. Masoomeh Yeganehjoo, who have inspired and continue to inspire education in students and people alike.

Chapter 1

Introduction

Social media is becoming a more prevalent part of our everyday life, due to the advancements in technology and virtualization. The availability of the Internet, cameras and real-time message boards at our fingertips has brought about live and parallel reporting, and witness testimonies during many events. These reports can be useful to responders and can help create awareness among the populace, especially in emergency situations (Meier, 2015; Watson et al., 2017). Despite the potential benefits, major response groups and organizations under-utilize these sources of information, as therein lie many administrative and technical challenges (Meier, 2013). Among the challenges, there are reliability issues associated with public and unstructured data, as well as information overload issues, as millions of messages are posted during a crisis situation (Bullock et al., 2012).

There are many recent studies that propose the use of machine learning techniques to provide automated methods for analyzing social media data to reduce the information overload (Imran et al., 2015; Beigi et al., 2016). Machine learning techniques can help transform raw data into usable information by labeling, prioritizing and structuring data, and making them beneficial to responders and to the populace in times of need (Qadir et al., 2016). However, supervised learning algorithms rely on labeled training data to build predictive models. Accurate labeling of data for an emerging disaster is both time consuming and expensive, and, hence, it is not appropriate to assume that labeled data for a current disaster will be

promptly available to be used for analysis. The lack of labeled data for emerging disasters prohibits the use of supervised learning techniques.

To address this challenge, several works proposed to use labeled data from prior "source" disasters to learn *supervised* classifiers for a "target" disaster (Verma et al., 2011; Imran et al., 2013, 2016b). However, due to the divergence of each disaster domain in terms of location, nature, season, etc. (Palen and Anderson, 2016), the source disaster might not accurately represent the characteristics of the target disaster (Qadir et al., 2016; Imran et al., 2015). Domain adaptation techniques (Pan and Yang, 2010; Jiang, 2008) are designed to circumvent the lack of labeled target data by making use of unlabeled target data as guideposts for the readily available labeled source data. Studies in the disaster space have shown that using domain adaptation techniques, which use together target unlabeled data and source labeled data, significantly improve classification results as compared to supervised techniques that use solely labeled source data (Li et al., 2015, 2017b). Unlabeled data from the target disaster become more abundant as the event unfolds, and it can enable the use of domain adaptation techniques during emerging or occurring disasters. This property of domain adaptation, which is in line with realistic data availability, makes it more appealing to employ, during emerging or occurring disasters.

There are several ways in which the unlabeled target data can be used with domain adaptation techniques, including parameter-based adaptation, instance-based adaptation and feature-based adaptation (Pan and Yang, 2010). In the parameter-based adaptation, the labeled source data is used together with the unlabeled target data to identify shared parameters that result in good predictions for the target data. In the instance-based adaptation, the unlabeled target data is used to identify and/or reweigh the most relevant source labeled instances with respect to the target classification task, while in feature-based adaptation, the target unlabeled data and source labeled data are used together to find a feature representation that minimizes the difference between the two domains. Prior work on disaster tweet classification using domain adaptation has relied on parameter-based adaptation. Specifically, Li et al. (2017b) proposed to learn weighted source and target Naïve Bayes classifiers with the iterative method of Expectation-Maximization (EM) (Dempster et al.,

1977), and showed that the resulting classifiers can accurately predict the target.

In this study, we propose to use a combination of two domain adaptation approaches, specifically a hybrid between feature-based adaptation and instance-based adaptation, to reduce the variation between the two domains. First, the Alternating Nonnegative Least Squares Matrix Factorization (LSNMF) (Lin, 2007) is used on the combined source and target data, represented using binary vectors, to create a dense and reduced conceptual representation of source and target instances. Subsequently, the k-Nearest Neighbors algorithm (kNN) is used to select a subset of the source instances which are most similar to the target instances, according to the cosine similarity calculated based on the reduced common representation. Data are represented using both binary (existence) and numeric (TF-IDF) representations, as these representations allow us to observe how the different feature types affect different method combinations. The objective is to gain an understanding of the benefits provided by the hybrid feature-instance adaptation approach, as compared to the independent feature or instance adaptation approaches. Furthermore, given that both the LSNMF approach and the kNN approach have parameters that need to be tuned, specifically, the number of reduced features f for LSNMF and the number of neighbors k for kNN, we aim to study the variation of performance with these parameters and identify overall good values that can be used in practice.

As an application, we focus on the task of classifying disaster tweets as being relevant to the disaster of interest (*i.e.*, on-topic) or not relevant (*i.e.*, off-topic). This is one of the most basic but crucial classifications needed during a disaster, as subsequent analysis should be done only on data relevant to the disaster in question. Furthermore, this classification is not trivial: supervised classifiers may not achieve accurate results due to domain variations. Hence, we use our feature-instance adaptation approach and perform comparisons to baselines such as supervised Naive Bayes and individual components of our approach (e.g., feature adaptation only or instance adaptation onclassification of the binary representationly). Furthermore, we compare our approach to an existing Self-Training Domain Adaptation approach, which does not perform feature or instance adaptation, but instead adapt parameters of the source model based on the target model.

To summarize, our main contributions are as follows:

- We design a hybrid feature-instance adaptation approach to adapt the source disaster data to the target disaster data. Specifically, we use a matrix factorization approach to construct a shared representation of source and target instances, and subsequently use the kNN algorithm to select source instances that are most similar to target instances. Finally, we train supervised Naïve Bayes classifiers on the modified source data.

- We perform an extensive set of experiments on pairs of source-target disasters from the CrisisLexT6 datasets to evaluate the feature-instance adaptation approach by comparison with approaches that make use of either feature-based adaptation or instance-based adaptation, but not both.

- We study the variation of performance with the parameters of the feature-based adaptation (specifically, the number of features, f), and instance-based adaptation (specifically, the number of neighbors, k), respectively, to identify parameters that result in good overall performance.

Chapter 2

Methods

There are many traditional machine learning techniques that can be used for disaster tweet classification, such as Naïve Bayes (NB), Support Vector Machines (SVM), Logistic Regression (LR), Random Forest (RF), etc. Compared to other algorithms, Naïve Bayes has the advantage of not requiring hyper-parameter tuning. Furthermore, a recent study on disaster tweet classification (Li et al., 2017b) has shown that the results obtained with Naïve Bayes are comparable, and sometimes better, than the results obtained with other more sophisticated algorithms used with default parameters. Therefore, in this work, we will use Naïve Bayes together with a hybrid feature-instance adaptation approach to learn classifiers for disaster data, as described below.

Given a source and target pair of disasters, our goal is to adapt the source data by reducing the variance with respect to the target data, and then train Naïve Bayes classifiers on the adapted source data. The source adaptation is guided by the target unlabeled data. More specifically, we propose a hybrid feature-instance adaptation approach to select a subset of the source instances, which are most similar to the target instances. First, the target instances are used to construct a target vocabulary V, which is subsequently used to represent both source and target data as bag-of-words binary vectors. As part of the feature adaptation step, the resulting data matrix D is decomposed using the popular Least Squares Nonnegative Matrix Factorization (LSNMF) proposed by Lin (2007). The implementation of this

method is available in Python under the "nimfa" package. Intuitively, the decomposition will produce a reduced dense representation of the data, which is more suitable for identifying similar instances as compared to the sparse binary representation (Guo and Diab, 2012).

As part of the instance adaptation step, the reduced representation is used to identify source instances that are most similar to the target. More precisely, for each target (unlabeled) instance, we calculate the cosine similarity to the source instances and select the k nearest neighbors from the source. If two different target instances have the same source instance among the k nearest neighbors, the selected subset of the source may contain duplicate instances. We experiment with two settings, one in which we retain duplicates (i.e., we reweigh source instances), and another one in which we remove duplicates (under the assumption that duplicates can bias the classifier).

Finally, we use the Naïve Bayes algorithm to learn classifiers from the selected subset of the source. Here, we also experiment with three settings: one in which the supervised Gaussian Naïve Bayes algorithm is used on the reduced representation of the selected source instances, and one in which the supervised Bernoulli Naïve Bayes algorithm is used on the original binary representation of the selected source instances, and another where we use a self-training domain adaptation technique based on a weighted Naïve Bays Bernoulli classifier proposed by Li et al. (2017a). The reason we also experiment with the binary representation of the adapted source is that in preliminary experimentation the binary representation gave better results than the numeric TF-IDF representation Finally, the resulting classifiers are tested on separate target test data. The approach is summarized in Algorithm 1.

Algorithm 1: Hybrid feature-instance adaptation with Naïve Bayes and domain adaptation classifiers

1. Given: Target unlabeled data TU, source labeled data SL, and target test data TT.
2. Use target unlabeled data TU to construct the vocabulary V.
3. Represent source SL and target TU data as binary vectors. The resulting data matrix is denoted by D.
4. *Feature adaptation*: Use the Least Squares Non-negative Matrix Factorization to obtain a reduced representation of the source and target data. The dimension of the reduced representation is denoted by f.
5. *Instance adaptation*: For each target instance in TU, find its k nearest neighbors and add them to the selected subset of source instances $Sel\text{-}SL$, by retaining duplicates or by removing duplicates, respectively.
6. *Naïve Bayes*: Use the selected subset of source instances $Sel\text{-}SL$, with the reduced representation or the original binary representation, respectively, to learn a classifier for the target data. Alternatively, use the Self-Training Domain Adaptation approach on the modified source/target.
7. Evaluate the resulting Naïve Bayes and Self-Training Domain Adaptation classifiers on the target test data TT.

Chapter 3

Dataset and Experimental Setup

3.1 Dataset

The CrisisLexT6 dataset (Olteanu et al., 2014) is a collection of six disasters that occurred between October 2012 and July 2013 in United States, Canada and Australia. This dataset was collected through Twitter API based on disaster keywords and the geographic locations of the affected areas. Each disaster's data contains approximately 10,000 tweets which were manually labeled as on-topic or off-topic using CrowdFlower, a popular crowdsourcing platform. The data was cleaned according to the pre-processing steps described in (Li et al., 2015), which included removing re-tweets (RT), duplicate tweets, non-printable ASCII characters, and replacing URL, email addresses and usernames with placeholders pertaining to each. Furthermore, the dataset is split into combinations of consecutive source-target pairs of all six disasters and converted into bag-of-words binary (word existence) representations. Each feature (word) must appear at least 10 times in any given pair of disasters to be included in the vocabulary as a feature. Hence, the feature set is different from one source-target pair to the another, although, on average, pairs have approximately 1200-1300 features.

3.2 Experimental Setup

In this section, we state the research questions that are driving our experiments, describe the evaluation setup and also the parameter setting for the constituent approaches of the experiments, and finally our baselines.

3.3 Research Questions

Our experiments are designed to answer the following research questions:

- Are the adaptation approaches more effective than the baseline, where Bernoulli Naïve Bayes is used to learn classifiers from the binary representation of the source data? Does this also hold true when classifying using the Self-Training Domain Adaptation approach? Are the same effects observed when using the numeric representation of the data?

- Is the hybrid feature-instance adaptation approach more effective than the individual feature adaptation and instance adaptation approaches? Between Gaussian Naïve Bayes on the reduced or numeric representation of the selected source data and Bernoulli Naïve Bayes on the binary representation of the selected numeric or binary source data, which classifier gives better results?

- Between the feature adaptation approach and the instance adaptation approach, which one is more effective? What parameter values result in better performance for the two approaches, respectively considering either the numerical or binary representation?

- When using the instance adaptation approach, is it better to keep duplicate neighbors or to remove them?

- How do the feature and hybrid approaches affect the Self-Training Domain Adaptation classifier?

Table 3.1: *Summary of source and target disaster pairs used in the experiments, together with information about instances and features in the combined source and target datasets*

Crisis			Instances			Features
Abbreviation	Source	Target	On-topic	Off-topic	Total	
BB-AF	Boston Bombings	Alberta Floods	7938	9023	16961	1322
BB-OT		Oklahoma Tornado	7650	9358	17008	1143
BB-WT		West Texas Explosion	8564	9042	17606	1239
OT-AF	Oklahoma Tornado	Alberta Floods	6706	9763	16469	1322
QF-AF	Queensland Floods	Alberta Floods	6733	9264	15997	1322
QF-BB		Boston Bombings	7677	8859	16536	1317
QF-OT		Oklahoma Tornado	6445	9599	16044	1143
SH-AF	Sandy Hurricane	Alberta Floods	8758	8466	17224	1322
SH-BB		Boston Bombings	9702	8061	17763	1317
SH-OT		Oklahoma Tornado	8470	8801	17271	1143
SH-QF		Queensland Floods	8497	8302	16799	1242
SH-WT		West Texas Explosion	9384	8485	17869	1239

Chapter 4

Evaluation Strategy and Baselines

4.1 Evaluation Strategy

We consider the six disasters in our dataset in chronological order and create 12 pairs of source and target disasters, by ensuring that the source disaster has occurred before the target disaster (under the assumption that a later disaster may mention an earlier disaster but not the other way around). This strategy creates pairs of natural or man-made disasters, but also pairs that contain a combination of natural and man-made disasters. In our result tables, we use the abbreviations shown in Table 3.1 to specify the source and target disasters in a pair, respectively. Out of the six disasters used as part of our dataset, BB (Boston Bombings) and WT (West Texas Explosion) are man-made while the others are natural disasters.

We used the 5-fold cross-validation technique for each pair of disasters to select target test and target unlabeled data. Similar to (Li et al., 2017b), the folds are rotated five times to obtain five combinations of consecutive folds, within each selecting the first three folds as target unlabeled TU data, the next fold as target test TT data, and last one as target labeled TL data (reserved for future work). Each domain has between 8000-9000 instances, as can be seen from Table 3.1. Only target unlabeled data is used with the instance adaptation approach, which means that the classifiers are different for each test fold, as they are trained

from different subsets of the source instances, as guided by the corresponding unlabeled target data. We report accuracy results averaged over 5 folds.

4.1.1 Matrix Factorization Setup

Data from each pair of disasters, represented as a binary matrix which consists of bag-of-words vectors, is reduced using the LSMNF technique. Specifically, the number of features f for each pair is reduced from approximately 1200-1300 to 30, 50, 100, 200, and 500 features, respectively.

4.1.2 K-Nearest Neighbors Setup

The kNN algorithm is used to select the k nearest neighbors from the entire source for each of the instances in the target unlabeled dataset. We experiment with the following values for k: 1, 3, 5, 7, 9, 11, to understand what value of k results in best overall performance. As there is a possibility of having the same source neighbor for multiple target instances, duplicates may exist in the source subset. Hence, we experiment with two options: retaining duplicates (d) or not retaining duplicates (n) to understand which one is more appropriate.

4.1.3 Bernoulli Naïve Bayes and Gaussian Naïve Bayes

After selecting a subset of the source instances using the hybrid feature-instance adaptation, the next step is to learn a Naïve Bayes classifier from the adapted source. We experiment with two options. First, we use the reduced representation of the selected source subset (r) to train Gaussian Naïve Bayes classifiers. Furthermore, we also use the original binary representation of the instances in the selected source subset (b) to train Bernoulli Naïve Bayes, given preliminary experimentation that showed better results with Bernoulli Naïve Bayes on the binary representation, as compared to Gaussian Naïve Bayes on the numeric representation.

4.1.4 Self-Training Domain Adaptation

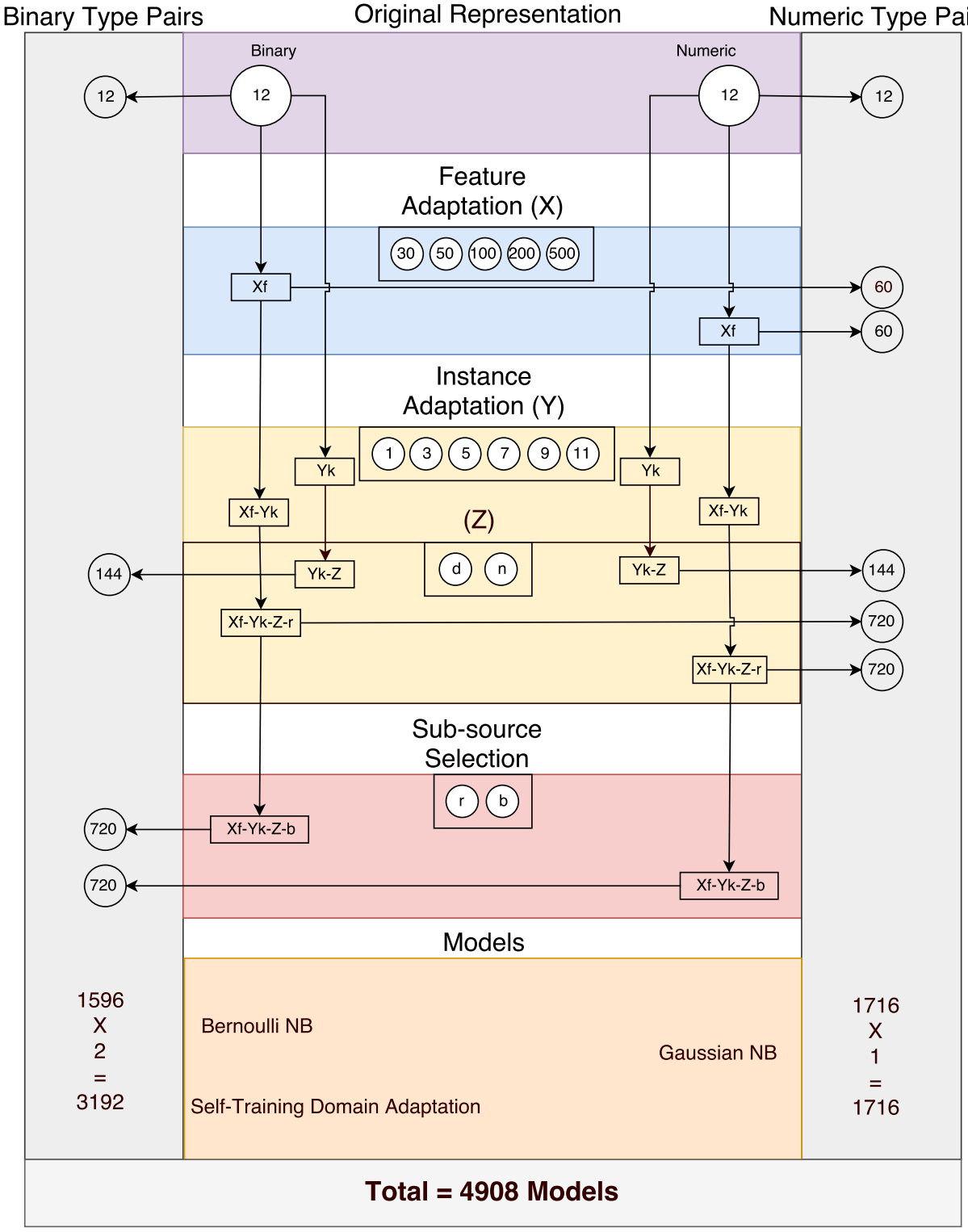
After reverting to the original binary feature set based on the source subset, we use a self-training domain adaptation classifier with Naïve Bayes Bernoulli as proposed in (Li et al., 2017a) to train a target model iteratively. This is done by performing a weighted combination of source and target until the predicted labels do not change significantly between two consecutive iterations. This model is trained and used to classify each of the pairs with binary features.

4.2 Total Number of Models Trained

Figure 4.1 provides a visual representation for this section. Initially, we consider the 12 original binary pairs of disasters, and then reduced each to the five feature sets mentioned in 4.1.1, resulting in 60 pairs of feature reduced datasets and corresponding models. Similarly, the numeric representation of the dataset will also create the same number of feature adapted pairs and models. Adding up the 24 original pairs (12 for the binary representation and 12 for the numeric representation), and the 60 feature adapted pairs, with different feature combinations. Furthermore, after performing the six combinations of the instance adaptation approach on the previous 60 pairs, creates 360 pairs of source subsets/models. This number doubles to 720 with the additional choice of removing duplicates in source subsets. That being said, the initial 60 pairs from feature adaptation should also be tested as a control for instance adaptation. Finally, by reverting to the original binary feature set mentioned in 4.1.3 for each pair we create another 720 pairs to model.

The Self-Training Domain Adaptation classifier is trained on the binary featured data, which is about half of the pairs, in addition to the Bernoulli Naïve Bays classifier, while the numeric featured data are only classified using the Gaussian Naïve Bays. Hence, we trained about 4908 models in this work.

Figure 4.1: *Hybrid workflow combinations and number of models trained considering the feature-instance domain adaptation approach in conjunction with the classification models. X is the number of reduced features, Y is the number of neighbors and Z is the duplication retaining policy. The previous stage of each adaptation approach is considered a baseline as to whether the baseline affects the accuracy of the trained models*



4.3 Baselines

We compare our proposed approach against the following baselines:

- *Supervised Bernoulli Naïve Bayes classifiers* learned from the binary representation of the source and evaluated on the test target data.
- *Supervised Gaussian Naïve Bayes classifiers* learned from the numeric representation of the source and evaluated on the test target data.
- *Self-Training Domain Adaptation classifiers* learned from the binary representation of the source and evaluated on the test target data.
- *Instance adaptation with Bernoulli Naïve Bayes classifiers*, where we first use the binary representation of the source to identify a subset of instances most similar to the target instances, and subsequently learn Bernoulli Naïve Bayes classifiers from the selected source subset.
- *Feature-adaptation with Gaussian Naïve Bayes classifiers*, where we first use the binary and numeric representations of the source and target to find a reduced dense representation, and subsequently learn Gaussian Naïve Bayes classifiers from the selected source subset.

Chapter 5

Experimental Results and Discussion

5.1 Instance Adaptation with Bernoulli Naïve Bayes Classifiers

Instance adaptation is performed on the original binary representation of the combined source and (unlabeled) target datasets using kNN. Specifically, for each target instance we select the k nearest neighbors from the corresponding source. Subsequently, Bernoulli Naïve Bayes is used on the selected source subset, with duplicates (d) or no duplicates (n). The goal of this adaptation is to subsample the portion of source which is closer to target, decreasing the variance between the two datasets. Table 5.1 shows the results of this set of experiments. As can be seen, the best results overall are obtained for the model labeled *3k-n* which is a model where the 3 nearest neighbors are selected for each target instance, and duplicates are not kept in the selected source subset. Furthermore, the performance slightly decreases for values of k greater than 3 (regardless of the fact that duplicates are retained or removed), suggesting that noisy source instances are added to the selected subset when more than 3 neighbors are included. Given this observation (and other preliminary experiments now shown), the subsequent experiments that make use of kNN will be run with $k = 3$ and $k = 7$ for binary and numeric represented pairs, respectively. When comparing the instance adaption results with the results of Bernoulli Naïve Bayes on the original binary data (labeled

Original in Table 5.1), it can be seen that the instance adaptation consistently improves the classification accuracy by as much as 7.4% and 6% in the case of SH-BB and BB-AF pairs, respectively. The same claim can be made when using Gaussian Naïve Bayes on the original numeric data (labeled *Original* in Table 5.2) with the exception of the SH-OT pair, observing improvements as high as 6.5% and 5.9% in the case of QF-OT and QF-BB pairs, respectively.

Table 5.1: *Instance-based adaptation using kNN on the binary representation, followed by Bernoulli Naïve Bayes, on the selected source subset. Accuracy results on 12 source-target pairs are shown for three values of k, specifically, 3, 5, and 7, and two instance-selection settings, specifically, with duplicates (denoted by d) and with no duplicates (denoted by n). For example, 3k-d means that 3 nearest neighbors are selected for each target instance, and duplicates are retained, while 3k-n means that 3 nearest neighbors are selected, but duplicates are removed. The Original results are obtained when Bernoulli Naïve Bayes is used on the original binary data. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.738 ±0.012	**0.843** ±0.013	**0.948** ±0.003	0.872 ±0.008	0.789 ±0.007	0.750 ±0.013	0.841 ±0.004	0.711 ±0.009	0.687 ±0.009	0.808 ±0.009	0.768 ±0.010	0.772 ±0.013
3k-d	**0.744** ±0.014	0.842 ±0.010	0.946 ±0.005	**0.871** ±0.005	**0.813** ±0.005	0.728 ±0.009	**0.848** ±0.008	**0.759** ±0.005	**0.755** ±0.007	**0.832** ±0.008	**0.823** ±0.006	0.820 ±0.019
5k-d	0.736 ±0.010	**0.847** ±0.009	**0.949** ±0.005	0.868 ±0.006	**0.811** ±0.005	0.717 ±0.007	**0.848** ±0.006	**0.756** ±0.010	**0.758** ±0.009	**0.830** ±0.006	**0.819** ±0.010	0.809 ±0.008
7k-d	0.732 ±0.008	0.844 ±0.008	**0.948** ±0.004	0.869 ±0.006	**0.810** ±0.004	0.712 ±0.006	**0.850** ±0.005	**0.757** ±0.007	**0.753** ±0.006	**0.828** ±0.008	0.816 ±0.012	0.830 ±0.006
3k-n	**0.752** ±0.015	**0.842** ±0.010	**0.946** ±0.004	**0.874** ±0.005	**0.806** ±0.005	**0.780** ±0.006	**0.853** ±0.010	0.726 ±0.009	**0.747** ±0.014	**0.829** ±0.008	0.789 ±0.010	**0.846** ±0.011
5k-n	**0.749** ±0.021	**0.846** ±0.010	**0.946** ±0.002	**0.874** ±0.007	0.804 ±0.005	0.770 ±0.005	0.850 ±0.008	0.724 ±0.008	0.739 ±0.013	0.823 ±0.007	0.786 ±0.014	0.833 ±0.008
7k-n	**0.746** ±0.017	**0.848** ±0.009	**0.947** ±0.003	**0.874** ±0.007	0.800 ±0.005	0.772 ±0.008	**0.849** ±0.008	0.720 ±0.005	0.733 ±0.021	0.817 ±0.009	0.782 ±0.011	0.820 ±0.012

5.2 Feature Adaptation with Gaussian Naïve Bayes Classifiers

Similar to the instance-based adaptation, the feature-based adaptation is also performed on the original binary and numeric data matrices, consisting of source and (unlabeled) target data. The goal of this adaptation is to create a denser feature set that better captures the

Table 5.2: *Instance-based adaptation using kNN on the numeric representation, followed by Gaussian Naïve Bayes, on the selected source subset. Accuracy results on 12 source-target pairs are shown for three values of k, specifically, 3, 5, and 7, and two instance-selection settings, specifically, with duplicates (denoted by d) and without duplicates (denoted by n). For example, 3k-d means that 3 nearest neighbors are selected for each target instance, and duplicates are retained, while 3k-n means that 3 nearest neighbors are selected, but duplicates are removed. The Original results are obtained when Gaussian Naïve Bayes is used on the original numeric data.The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.561 ±0.015	0.789 ±0.010	**0.914** ±0.007	0.720 ±0.006	0.612 ±0.009	0.449 ±0.017	0.573 ±0.009	**0.760** ±0.009	0.631 ±0.015	**0.777** ±0.008	**0.799** ±0.005	0.791 ±0.007
3k-d	**0.617** ±0.023	0.691 ±0.018	0.882 ±0.011	0.733 ±0.014	0.618 ±0.017	**0.508** ±0.022	**0.631** ±0.012	0.734 ±0.013	0.661 ±0.020	0.731 ±0.007	0.768 ±0.014	0.820 ±0.008
5k-d	0.598 ±0.011	0.722 ±0.012	0.899 ±0.007	**0.749** ±0.020	**0.630** ±0.014	**0.505** ±0.025	**0.638** ±0.010	0.744 ±0.011	**0.677** ±0.022	0.740 ±0.008	0.782 ±0.009	0.829 ±0.004
7k-d	**0.603** ±0.009	**0.749** ±0.006	0.908 ±0.006	**0.757** ±0.011	**0.637** ±0.012	0.490 ±0.025	**0.635** ±0.013	0.746 ±0.011	0.654 ±0.019	0.752 ±0.005	**0.787** ±0.010	0.826 ±0.006
3k-n	0.597 ±0.029	0.648 ±0.017	0.890 ±0.011	0.687 ±0.021	0.574 ±0.017	0.469 ±0.016	0.562 ±0.008	0.749 ±0.016	0.657 ±0.023	0.750 ±0.006	0.780 ±0.008	**0.837** ±0.006
5k-n	0.567 ±0.014	0.696 ±0.012	0.909 ±0.009	0.696 ±0.031	0.593 ±0.020	0.462 ±0.016	0.585 ±0.014	**0.764** ±0.012	0.648 ±0.024	0.758 ±0.006	**0.797** ±0.008	**0.837** ±0.008
7k-n	0.570 ±0.016	**0.741** ±0.011	**0.919** ±0.009	0.705 ±0.025	0.601 ±0.014	0.445 ±0.015	0.589 ±0.010	**0.757** ±0.017	0.649 ±0.026	0.763 ±0.003	**0.791** ±0.007	0.808 ±0.018

similarity between target and source instances, and ultimately produces better classification results. We use a wide range of dimensions, specifically 30, 50, 100, 200 and 500. Table 5.3 shows the results of the Gaussian Naïve Bayes classifiers trained on the reduced representations from the original binary representation, by comparison with the results of the Bernoulli Naïve Bayes classifiers trained on the original binary representation. Similarly, Table 5.4 shows results with the minor difference of using reduced representations from the numeric representation and the Gaussian Naïve Bayes classifier on the original numeric representation.

As can be seen, the highest accuracy overall is obtained with the reduced representation, although there are pairs in Table 5.3 for which the original representation gives better results. This suggests that the reduced representation by itself is not always enough to ensure best results on the target. We hypothesize that the reason behind the highest values obtained with the original models could be given by the overall similarity of those source and target disasters, as reflected by the higher accuracy scores when compared with other original models. It can also be observed that three of the four cases in Table 5.3 where feature adaptation did not perform well, OT is one of the disasters in the pair, which might hint to an anomaly in this specific disaster. The results in Table 5.3 also show that the classifiers trained with 200 reduced features (i.e., *200f*) give the best results overall, while sometimes the models trained with 50 or 100 reduced features give the best results for specific pairs. Table 5.4 results show that models trained with 30 reduced features give the overall best result, while the ones trained with 200 perform well when the 30 feature model falls short. In subsequent experiments we will only train classifiers with 50 and 200 features for pairs from the binary representation, and 30 and 200 features for pairs from the numeric representation to reduce the number of experiments (by eliminating several values from the original feature adaptation experiment).

Table 5.3: *Feature-based adaptation using LSNMF on the original binary representation, followed by Gaussian Naïve Bayes on the reduced representation. Accuracy results on 12 source-target pairs are shown for six values of f, specifically, 30, 50, 100, 200, and 500. For example, $50f$ means that the LSNMF decomposition has 50 reduced features. The Original results are obtained with Bernoulli Naïve Bayes on the original binary data. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.738 ±0.012	**0.843** ±0.013	**0.948** ±0.003	**0.872** ±0.008	0.789 ±0.007	**0.750** ±0.013	**0.841** ±0.004	0.711 ±0.009	0.687 ±0.009	**0.808** ±0.009	0.768 ±0.010	0.772 ±0.013
30f	**0.792** ±0.010	0.801 ±0.013	0.922 ±0.004	0.756 ±0.011	**0.860** ±0.007	0.444 ±0.009	0.770 ±0.011	0.720 ±0.008	**0.807** ±0.005	0.751 ±0.009	0.781 ±0.004	0.704 ±0.012
50f	0.764 ±0.011	**0.850** ±0.010	0.921 ±0.006	0.774 ±0.002	0.796 ±0.006	0.457 ±0.009	0.760 ±0.013	0.790 ±0.005	0.565 ±0.015	0.766 ±0.011	0.809 ±0.013	0.736 ±0.002
100f	0.563 ±0.022	0.829 ±0.010	**0.948** ±0.005	0.798 ±0.006	0.849 ±0.006	0.643 ±0.011	0.808 ±0.012	**0.828** ±0.008	0.698 ±0.010	0.758 ±0.009	0.819 ±0.007	**0.843** ±0.016
200f	0.618 ±0.012	**0.851** ±0.009	0.932 ±0.004	0.814 ±0.007	0.841 ±0.010	0.729 ±0.018	0.815 ±0.007	**0.834** ±0.003	0.669 ±0.010	0.743 ±0.011	**0.833** ±0.002	**0.846** ±0.013
500f	0.721 ±0.013	0.807 ±0.011	0.936 ±0.003	0.815 ±0.006	0.824 ±0.007	0.463 ±0.012	0.694 ±0.007	0.799 ±0.012	0.671 ±0.007	0.742 ±0.009	**0.840** ±0.007	0.825 ±0.011

Table 5.4: *Feature-based adaptation using LSNMF on the original numeric representation, followed by Gaussian Naïve Bayes on the reduced representation. Accuracy results on 12 source-target pairs are shown for six values of f, specifically, 30, 50, 100, 200, and 500. For example, $50f$ means that the LSNMF decomposition has 50 reduced features. The Original results are obtained with Gaussian Naïve Bayes on the original numeric data. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.561 ±0.015	0.789 ±0.010	0.914 ±0.007	0.720 ±0.006	0.612 ±0.009	0.449 ±0.017	0.573 ±0.009	0.760 ±0.009	0.631 ±0.015	0.777 ±0.008	0.799 ±0.005	0.791 ±0.007
30f	0.768 ±0.012	0.791 ±0.011	0.891 ±0.007	0.776 ±0.012	0.835 ±0.004	**0.778** ±0.008	0.797 ±0.011	0.774 ±0.010	**0.805** ±0.008	**0.851** ±0.009	**0.836** ±0.009	**0.886** ±0.007
50f	**0.842** ±0.007	0.800 ±0.011	0.913 ±0.005	0.777 ±0.014	**0.846** ±0.003	0.761 ±0.012	0.781 ±0.014	0.789 ±0.012	**0.802** ±0.005	0.835 ±0.007	0.820 ±0.015	0.873 ±0.009
100f	0.813 ±0.009	0.794 ±0.003	0.916 ±0.006	0.811 ±0.009	**0.831** ±0.005	0.759 ±0.010	**0.817** ±0.014	**0.823** ±0.008	0.785 ±0.011	0.808 ±0.012	0.828 ±0.007	0.864 ±0.008
200f	0.809 ±0.010	0.798 ±0.010	**0.934** ±0.008	**0.823** ±0.005	0.795 ±0.012	0.695 ±0.020	0.800 ±0.014	**0.818** ±0.007	0.753 ±0.008	0.795 ±0.006	0.817 ±0.003	0.862 ±0.010
500f	0.605 ±0.008	**0.821** ±0.012	**0.931** ±0.005	0.804 ±0.002	0.703 ±0.007	0.542 ±0.014	0.699 ±0.013	0.770 ±0.015	0.649 ±0.006	0.727 ±0.008	0.819 ±0.004	0.760 ±0.008

Table 5.5: *Hybrid feature-instance adaptation. Accuracy results on 12 source-target pairs from the binary representation are shown for $k = 3$, $f = 50, 200$, respectively, combined with settings with duplicates (denoted by d) or with no duplicates (denoted by n). Naïve Bayes is run on the selected source subset with reduced (denoted by r) and binary (denoted by b) representations, respectively. For example, 50f-3k-d-r means that LSNMF gives* 50 *reduced features, kNN selects 3 nearest neighbors, duplicated are retained, and the Gaussian Naïve Bayes is trained on the reduced representation, while 50f-3k-n-b means that there are no duplicates and Bernoulli Naïve Bayes is trained on the binary representation of the selected source subset. The Original results are obtained when Bernoulli Naïve Bayes is used on the original data. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.738 ±0.012	**0.843** ±0.013	**0.948** ±0.003	**0.872** ±0.008	0.789 ±0.007	0.750 ±0.013	0.841 ±0.004	0.711 ±0.009	0.687 ±0.009	0.808 ±0.009	0.768 ±0.010	0.772 ±0.013
50f-3k-d-r	0.757 ±0.012	0.822 ±0.009	0.928 ±0.008	0.753 ±0.007	0.780 ±0.007	0.749 ±0.009	0.758 ±0.012	0.775 ±0.010	0.645 ±0.099	0.696 ±0.012	0.796 ±0.007	0.883 ±0.006
50f-3k-n-r	0.742 ±0.009	0.828 ±0.007	0.934 ±0.007	0.771 ±0.005	0.775 ±0.004	0.618 ±0.013	0.758 ±0.012	0.757 ±0.010	0.653 ±0.105	0.757 ±0.007	0.765 ±0.008	**0.884** ±0.019
200f-3k-d-r	0.705 ±0.016	0.816 ±0.011	0.923 ±0.010	0.799 ±0.007	**0.834** ±0.008	**0.771** ±0.024	0.803 ±0.009	**0.838** ±0.008	**0.797** ±0.014	0.707 ±0.012	0.833 ±0.006	0.828 ±0.059
200f-3k-n-r	0.669 ±0.015	0.789 ±0.010	0.931 ±0.008	0.815 ±0.007	0.820 ±0.011	0.762 ±0.013	0.788 ±0.011	0.803 ±0.012	0.753 ±0.030	0.691 ±0.020	0.790 ±0.005	0.806 ±0.065
50f-3k-d-b	**0.782** ±0.004	**0.846** ±0.007	0.940 ±0.006	0.868 ±0.006	0.810 ±0.005	0.716 ±0.017	**0.848** ±0.005	0.805 ±0.014	0.746 ±0.013	0.815 ±0.005	**0.851** ±0.008	**0.895** ±0.006
50f-3k-n-b	0.764 ±0.008	**0.840** ±0.011	**0.945** ±0.005	**0.873** ±0.008	0.807 ±0.005	**0.773** ±0.011	**0.850** ±0.007	0.773 ±0.011	0.762 ±0.025	**0.846** ±0.009	0.834 ±0.004	**0.891** ±0.004
200f-3k-d-b	0.758 ±0.010	0.836 ±0.010	0.926 ±0.009	0.865 ±0.005	0.773 ±0.006	0.694 ±0.011	0.822 ±0.002	0.789 ±0.007	0.766 ±0.007	0.815 ±0.006	**0.858** ±0.003	0.868 ±0.012
200f-3k-n-b	0.766 ±0.013	0.830 ±0.009	0.939 ±0.008	**0.874** ±0.006	0.795 ±0.008	**0.762** ±0.008	0.831 ±0.003	0.784 ±0.008	0.776 ±0.004	**0.843** ±0.004	0.839 ±0.010	**0.897** ±0.010

5.3 Hybrid Feature-Instance Adaptation with Bernoulli or Gaussian Naïve Bayes

Finally, we experiment with our proposed hybrid feature-instance adaptation approach combined with Gaussian and Bernoulli Naïve Bayes classifiers, respectively. We fix the value of k 3 and 7 for pairs from the binary and numeric representations, respectively, as each value gave the best results in our instance adaptation experiments. We also fix f to 50 or 200 reduced features for the binary representation, and to 30 or 200 reduced features for the numeric representation, respectively. For kNN, we experiment with duplicates (d) and with

Table 5.6: *Hybrid feature-instance adaptation. Accuracy results on 12 source-target pairs from the numeric representation are shown for $k = 7$, $f = 30, 200$, respectively, combined with settings with duplicates (denoted by d) or with no duplicates (denoted by n). Naïve Bayes is run on the selected source subset with reduced (denoted by r) and binary (denoted by b) representations, respectively. For example, 30f-7k-d-r means that LSNMF gives* 30 *reduced features, kNN selects 7 nearest neighbors, duplicated are retained, and the Gaussian Naïve Bayes is trained on the reduced representation, while 50f-3k-n-b means that there are no duplicates and Bernoulli Naïve Bayes is trained on the binary representation of the selected source subset. The Original results are obtained when Gaussian Naïve Bayes is used on the original data. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.561 ±0.015	0.789 ±0.010	0.914 ±0.007	0.449 ±0.017	0.612 ±0.009	0.449 ±0.017	0.573 ±0.009	0.760 ±0.009	0.631 ±0.015	0.777 ±0.008	0.799 ±0.005	0.791 ±0.007
30f-7k-d-r	0.719 ±0.010	0.771 ±0.010	0.913 ±0.005	0.775 ±0.010	**0.842** ±0.008	**0.773** ±0.011	0.821 ±0.009	0.727 ±0.010	0.775 ±0.013	0.783 ±0.022	0.808 ±0.011	0.847 ±0.018
30f-7k-n-r	0.759 ±0.010	0.785 ±0.007	0.892 ±0.006	0.776 ±0.013	**0.847** ±0.004	**0.789** ±0.006	0.811 ±0.012	0.697 ±0.007	**0.807** ±0.009	**0.848** ±0.011	0.839 ±0.011	**0.883** ±0.008
200f-7k-d-r	**0.807** ±0.015	0.754 ±0.013	0.881 ±0.007	0.787 ±0.003	0.803 ±0.011	0.723 ±0.019	0.752 ±0.017	0.774 ±0.011	0.784 ±0.009	0.749 ±0.011	0.813 ±0.006	0.844 ±0.008
200f-7k-n-r	0.775 ±0.014	0.729 ±0.016	0.930 ±0.006	0.820 ±0.007	0.802 ±0.011	0.694 ±0.022	0.745 ±0.018	0.742 ±0.013	0.747 ±0.008	0.748 ±0.006	0.756 ±0.011	0.798 ±0.006
30f-7k-d-b	0.770 ±0.010	**0.845** ±0.006	0.924 ±0.005	**0.868** ±0.005	0.803 ±0.008	0.694 ±0.016	**0.849** ±0.004	**0.815** ±0.012	0.718 ±0.010	0.816 ±0.007	**0.868** ±0.009	0.842 ±0.004
30f-7k-n-b	0.771 ±0.010	**0.840** ±0.010	**0.954** ±0.003	**0.872** ±0.009	0.824 ±0.007	**0.784** ±0.009	**0.852** ±0.007	0.767 ±0.008	0.725 ±0.016	**0.854** ±0.008	0.833 ±0.012	0.860 ±0.012
200f-7k-d-b	0.757 ±0.011	**0.837** ±0.012	0.909 ±0.009	0.859 ±0.006	0.816 ±0.016	0.653 ±0.016	0.813 ±0.004	0.787 ±0.005	0.707 ±0.012	0.796 ±0.006	**0.861** ±0.006	0.806 ±0.014
200f-7k-n-b	0.762 ±0.013	0.835 ±0.007	0.945 ±0.005	**0.872** ±0.008	0.752 ±0.005	0.760 ±0.006	0.828 ±0.004	0.771 ±0.008	0.763 ±0.006	**0.855** ±0.007	0.828 ±0.013	**0.869** ±0.015

no-duplicates (n) options. Finally, once we select a subset of the source, we train Gaussian Naïve Bayes classifiers on the reduced representation of that subset (r), and Bernoulli Naïve Bayes classifiers on the binary representation of that subset (b). The results of the experiments are shown in Tables 5.5 and 5.6. As can be seen, the results of the hybrid approach are overall better than the results of the original models.

In Table 5.5 specifically, SH-AF and SH-WT, the increase in performance is close to 13%. Between duplicates and no-duplicates options, the no-duplicates option is usually better than the duplicates option, suggesting that the combination of feature and instance adaptation is good at identifying source instances that are representative for the target and prevents the need for changing the weights of the source instances (which was already apparent in the instance adaptation approach that used the sparse binary representation to find neighbors). Regarding the number of reduced features f, the results obtained with 50 features are overall better than the results obtained with 200 features. However, when looking at duplicate retainment and feature reduction together, we observe that they affect each other. Table 5.6 shows increases in performance of about 42% on OT-AF and 28% on QF-OT pairs.

For example, in Table 5.5 we can compare the difference between *50f-3k-d-b* and *50-3k-n-b*, on one hand, and *200f-3k-d-b* and *200-3k-n-b*, on the other hand. It can be observed that in the case of *50f* features the performance is overall higher for the no-duplicates option, as compared to the duplicates option, while this is not the case when considering *200f* features. Intuitively, a higher-level representation (i.e., smaller number of features) helps identify good nearest neighbors, which in turn helps obtain good performance. We could also gain this intuition from Table 5.6.

Finally, when comparing the performance of the Gaussian Naïve Bayes classifiers with the performance of the Bernoulli Naïve Bayes classifiers, the results are not conclusive: Gaussian Naïve Bayes classifiers give better results for half of the pairs, while Bernoulli Naïve Bayes classifiers give better results for the other half.

Chapter 6

Summary of the Results and Discussion

A summary of our results is shown in Tables 6.1 and 6.2, where we compare the original classifiers with the feature adaptation, instance adaptation and hybrid feature-instance adaptation classifiers. We will use the results in this table to answer our original research questions.

Are the adaptation approaches more effective than the baseline, where Bernoulli Naïve Bayes is used to learn classifiers from the binary representation of the source data? Are the same effects observed when using the numeric representation of the data? As can be seen from Tables 6.1 and 6.2, the adaptation-based classifiers are generally significantly better than the original classifiers.

Is the hybrid feature-instance adaptation approach more effective than the individual feature adaptation and instance adaptation approaches? Between Gaussian Naïve Bayes on the reduced or numeric representation of the selected source data and Bernoulli Naïve Bayes on the binary representation of the selected source data from either the reduced from the initial binary or numeric representations, which classifier gives better results? This question can be answered using Figures 6.1 and 6.2 which show the mean accuracy over all pairs and values of k for instance adaptation. The two figures show that on the average, regardless of

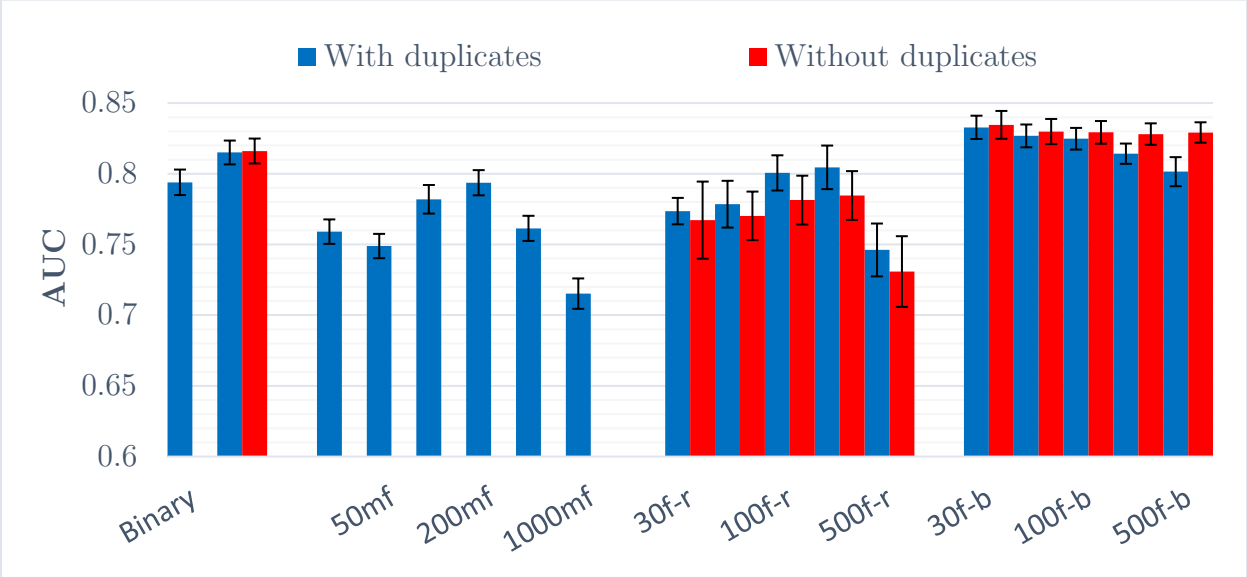
Table 6.1: *Summary of the results for 12 source-target pairs originating from the binary representation. The upper section of the table contains the individual feature adaptation (50f and 200f) and instance adaptation (3k-d and 3k-n) approaches, while the bottom section contains the hybrid approach for 50f and 200f, respectively, and 3k. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.738 ±0.012	**0.843** ±0.013	**0.948** ±0.003	**0.872** ±0.008	0.789 ±0.007	0.750 ±0.013	0.841 ±0.004	0.711 ±0.009	0.687 ±0.009	0.808 ±0.009	0.768 ±0.010	0.772 ±0.013
3k-d	0.744 ±0.014	0.842 ±0.010	**0.946** ±0.005	0.871 ±0.005	0.813 ±0.005	0.728 ±0.009	**0.848** ±0.008	0.759 ±0.005	0.755 ±0.007	0.832 ±0.008	0.823 ±0.006	0.820 ±0.019
3k-n	0.752 ±0.015	0.842 ±0.010	**0.946** ±0.004	**0.874** ±0.005	0.806 ±0.005	**0.780** ±0.006	**0.853** ±0.010	0.726 ±0.009	0.747 ±0.014	0.829 ±0.008	0.789 ±0.010	0.846 ±0.011
50f	0.764 ±0.011	**0.850** ±0.010	0.921 ±0.006	0.774 ±0.002	0.796 ±0.006	0.457 ±0.009	0.760 ±0.013	0.790 ±0.005	0.565 ±0.015	0.766 ±0.011	0.809 ±0.013	0.736 ±0.002
200f	0.618 ±0.012	**0.851** ±0.009	0.932 ±0.004	0.814 ±0.007	**0.841** ±0.010	0.729 ±0.018	0.815 ±0.007	**0.834** ±0.003	0.669 ±0.010	0.743 ±0.011	0.833 ±0.002	0.846 ±0.013
50f-3k-d-r	0.757 ±0.012	0.822 ±0.009	0.928 ±0.008	0.753 ±0.007	0.780 ±0.007	0.749 ±0.009	0.758 ±0.012	0.775 ±0.010	0.645 ±0.099	0.696 ±0.012	0.796 ±0.007	0.883 ±0.006
50f-3k-n-r	0.742 ±0.009	0.828 ±0.007	0.934 ±0.007	0.771 ±0.005	0.775 ±0.004	0.618 ±0.013	0.758 ±0.012	0.757 ±0.010	0.653 ±0.105	0.757 ±0.007	0.765 ±0.008	**0.884** ±0.019
50f-3k-d-b	**0.782** ±0.004	**0.846** ±0.007	0.940 ±0.006	0.868 ±0.006	0.810 ±0.005	0.716 ±0.017	**0.848** ±0.005	0.805 ±0.014	0.746 ±0.013	0.815 ±0.005	**0.851** ±0.008	**0.895** ±0.006
50f-3k-n-b	0.764 ±0.008	0.840 ±0.011	**0.945** ±0.005	**0.873** ±0.008	0.807 ±0.005	**0.773** ±0.011	**0.850** ±0.007	0.773 ±0.011	**0.762** ±0.025	**0.846** ±0.009	0.834 ±0.004	**0.891** ±0.004
200f-3k-d-r	0.705 ±0.016	0.816 ±0.011	0.923 ±0.010	0.799 ±0.007	**0.834** ±0.008	**0.771** ±0.024	0.803 ±0.009	**0.838** ±0.008	**0.797** ±0.014	0.707 ±0.012	0.833 ±0.006	**0.828** ±0.059
200f-3k-n-r	0.669 ±0.015	0.789 ±0.010	0.931 ±0.008	0.815 ±0.007	0.820 ±0.011	0.762 ±0.013	0.788 ±0.011	0.803 ±0.012	0.753 ±0.030	0.691 ±0.020	0.790 ±0.005	0.806 ±0.065
200f-3k-d-b	0.758 ±0.010	0.836 ±0.010	0.926 ±0.009	0.865 ±0.005	0.773 ±0.006	0.694 ±0.011	0.822 ±0.002	0.789 ±0.007	0.766 ±0.007	0.815 ±0.006	**0.858** ±0.003	0.868 ±0.012
200f-3k-n-b	0.766 ±0.013	0.830 ±0.009	0.939 ±0.008	**0.874** ±0.006	0.795 ±0.008	0.762 ±0.008	0.831 ±0.003	0.784 ±0.008	0.776 ±0.004	**0.843** ±0.004	0.839 ±0.010	**0.897** ±0.010

Table 6.2: *Summary of the results for 12 source-target pairs originating from the numeric representation. The upper section of the table contains the individual feature adaptation (30f and 200f) and instance adaptation (7k-d and 7k-n) approaches, while the bottom section contains the hybrid approach for 30f and 200f, respectively, and 7k. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.561 ±0.015	0.789 ±0.010	0.914 ±0.007	0.449 ±0.017	0.612 ±0.009	0.449 ±0.017	0.573 ±0.009	0.760 ±0.009	0.631 ±0.015	0.777 ±0.008	0.799 ±0.005	0.791 ±0.007
7k-d	0.603 ±0.009	0.749 ±0.006	0.908 ±0.006	0.757 ±0.011	0.637 ±0.012	0.490 ±0.025	0.635 ±0.013	0.746 ±0.011	0.654 ±0.019	0.752 ±0.005	0.787 ±0.010	0.826 ±0.006
7k-n	0.570 ±0.016	0.741 ±0.011	0.919 ±0.009	0.705 ±0.025	0.601 ±0.014	0.445 ±0.015	0.589 ±0.010	0.757 ±0.017	0.649 ±0.026	0.763 ±0.003	0.791 ±0.007	0.808 ±0.018
30f	0.768 ±0.012	0.791 ±0.011	0.891 ±0.007	0.776 ±0.012	0.835 ±0.004	**0.778** ±0.008	0.797 ±0.011	0.774 ±0.010	**0.805** ±0.008	**0.851** ±0.009	0.836 ±0.009	**0.886** ±0.007
200f	**0.809** ±0.010	0.798 ±0.010	0.934 ±0.008	0.823 ±0.005	0.795 ±0.012	0.695 ±0.020	0.800 ±0.014	**0.818** ±0.007	0.753 ±0.008	0.795 ±0.006	0.817 ±0.003	0.862 ±0.010
30f-7k-d-r	0.719 ±0.010	0.771 ±0.010	0.913 ±0.005	0.775 ±0.010	**0.842** ±0.008	**0.773** ±0.011	0.821 ±0.009	0.727 ±0.010	0.775 ±0.013	0.783 ±0.022	0.808 ±0.011	0.847 ±0.018
30f-7k-n-r	0.759 ±0.010	0.785 ±0.007	0.892 ±0.006	0.776 ±0.013	**0.847** ±0.004	**0.789** ±0.006	0.811 ±0.012	0.697 ±0.007	**0.807** ±0.009	**0.848** ±0.011	0.839 ±0.011	**0.883** ±0.008
30f-7k-d-b	0.770 ±0.010	**0.845** ±0.006	0.924 ±0.005	**0.868** ±0.005	0.803 ±0.008	0.694 ±0.016	**0.849** ±0.004	**0.815** ±0.012	0.718 ±0.010	0.816 ±0.007	**0.868** ±0.009	0.842 ±0.004
30f-7k-n-b	0.771 ±0.010	**0.840** ±0.010	**0.954** ±0.003	**0.872** ±0.009	0.824 ±0.007	**0.784** ±0.009	**0.852** ±0.007	0.767 ±0.008	0.725 ±0.016	**0.854** ±0.008	0.833 ±0.012	0.860 ±0.012
200f-7k-d-r	**0.807** ±0.015	0.754 ±0.013	0.881 ±0.007	0.787 ±0.003	0.803 ±0.011	0.723 ±0.019	0.752 ±0.017	0.774 ±0.011	0.784 ±0.009	0.749 ±0.011	0.813 ±0.006	0.844 ±0.008
200f-7k-n-r	0.775 ±0.014	0.729 ±0.016	0.930 ±0.006	0.820 ±0.007	0.802 ±0.011	0.694 ±0.022	0.745 ±0.018	0.742 ±0.013	0.747 ±0.008	0.748 ±0.006	0.756 ±0.011	0.798 ±0.006
200f-7k-d-b	0.757 ±0.011	**0.837** ±0.012	0.909 ±0.009	0.859 ±0.006	0.816 ±0.016	0.653 ±0.016	0.813 ±0.004	0.787 ±0.005	0.707 ±0.012	0.796 ±0.006	**0.861** ±0.006	0.806 ±0.014
200f-7k-n-b	0.762 ±0.013	0.835 ±0.007	0.945 ±0.005	**0.872** ±0.008	0.752 ±0.005	0.760 ±0.006	0.828 ±0.004	0.771 ±0.008	0.763 ±0.006	**0.855** ±0.007	0.828 ±0.013	**0.869** ±0.015

Figure 6.1: *Mean for 12 binary source-target pairs tested on k odd values from 1-11 for each approach. This figure compares the original Binary mean with the instance adaptation approach k, feature adaptation approach 30-500mf, and the hybrid approaches of the reduced r and binary feature approach b using different numbers of features. The effect of retaining and removing instances are also shown as adjacent values in approaches that include the instance adaptation process. The error bars show the confidence interval of each model's accuracy.*

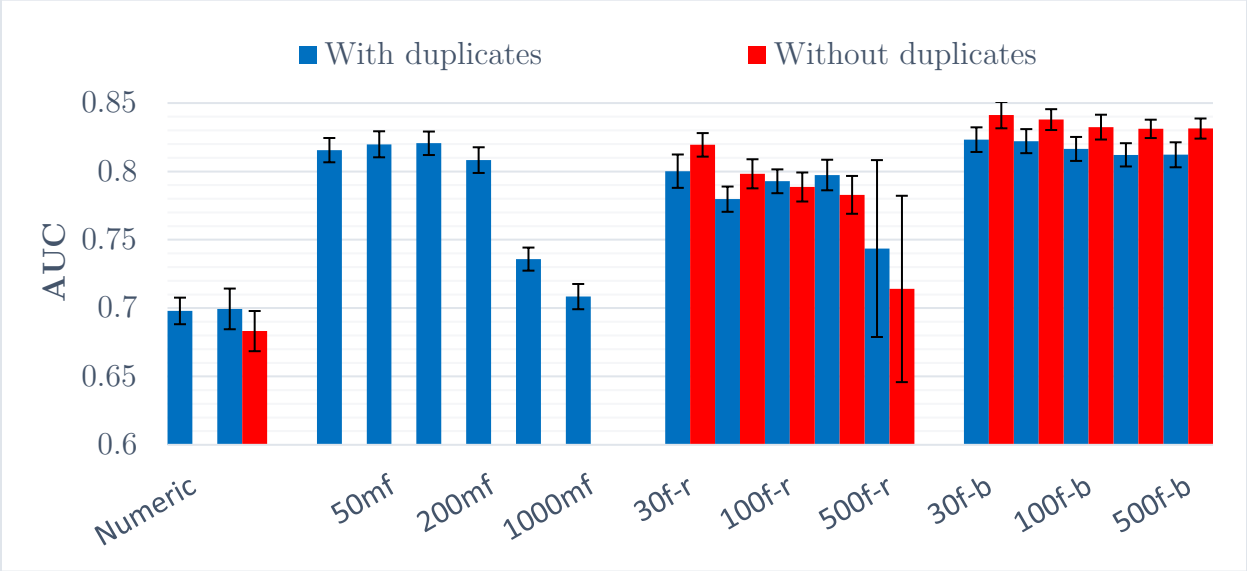


the number of reduced features, the hybrid approach maximizes accuracy especially when duplicates are not retained. We can also observe a significant increase of accuracy in Figure 6.2 from the original *Numeric* features compared to feature reduced or hybrid approach.

The question above can also be answered more specifically based on Tables 6.1 and 6.2. We have separated each table into two sections: one for the individual feature adaptation and instance adaptation approaches, and the other one for the hybrid approach. The best results for each pair (based on a t-test with $p \leq 0.05$) are highlighted in bold.

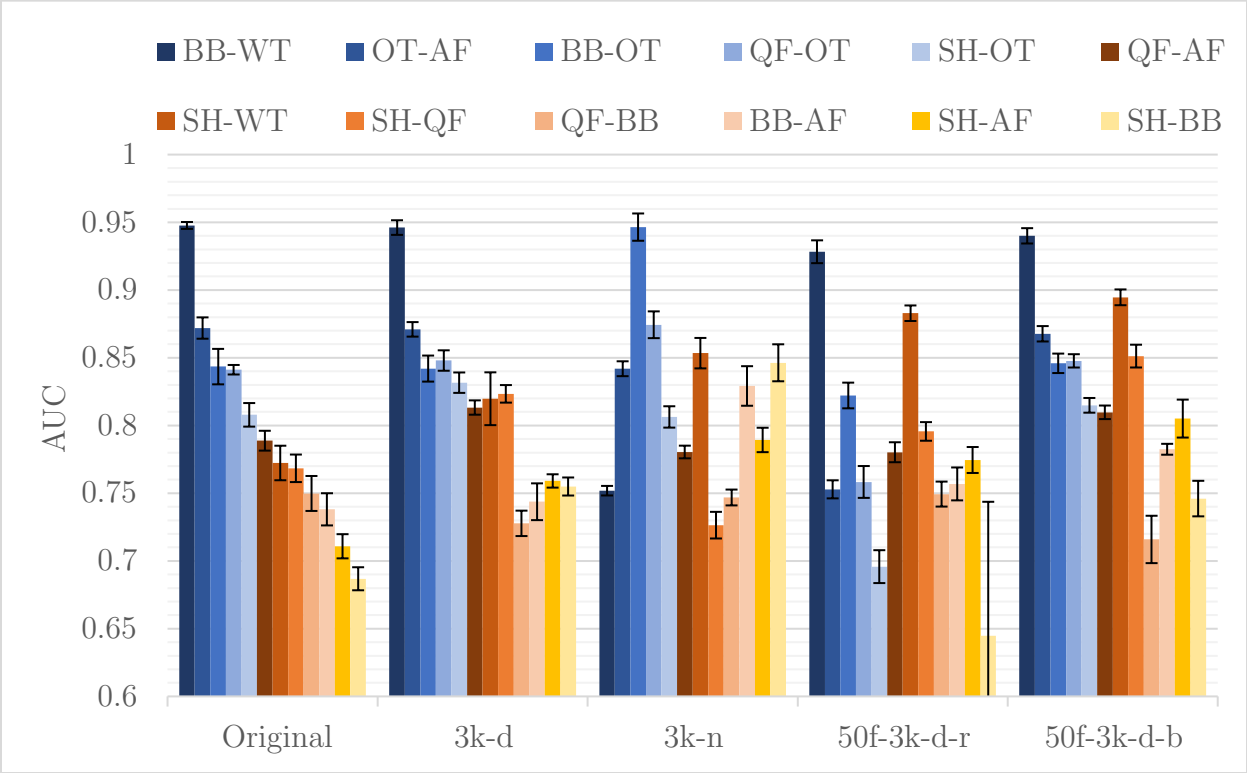
As can be seen in Table 6.1, when using the original binary representation, the hybrid approach achieves best results for all 12 pairs, while the feature adaptation approach achieves best results for only 3 pairs, and the instance adaptation approach achieves best results for only 4 pairs. While the individual adaptation approaches with *200f* and *3k-n* achieve best results for 7 pairs combined, the other results obtained with these approaches are not competitive. The hybrid approach with *50f-3k-d-b* and *50f-3k-n-b* settings achieves either

Figure 6.2: *Mean for 12 numeric source-target pairs tested on k odd values from 1-11 for each approach. This figure compares the original Numeric mean with the instance adaptation approach k, feature adaptation approach 30-500mf and the hybrid approaches of the reduced r and binary feature approach b using different numbers of features. The effect of retaining and removing duplicate instances are also shown as adjacent values in approaches that include the instance adaptation process. The error bars show the confidence interval of each model's accuracy.*



best values for almost all pairs or values closest to the best values for other pairs. In other words, the individual adaptation approaches can produce very good results in some cases and poor results in other cases, while the hybrid feature-instance adaptation approach with *50f, 3k* and no duplicates can produce competitive results consistently, suggesting that this approach is more reliable. Table 6.2, where instances are represented using the original numeric features, also shows that the hybrid approach gives the best results for all pairs. However, in half of the the pairs, the feature adaptation approach is able to match the beset results of the hybrid approach. Similar to table 6.1, the hybrid approach with the lower initially reduced feature set selecting from the original feature set, *30-7k-d-b* and *30-7k-n-b*, achieve the best result or values close to it. Moreover, the hybrid approach which does not retain duplicates seems to perform better than the one which retains duplicates, similar to what was discovered using the original binary features.

Figure 6.3: *Displaying the change in the accuracy of each pair of binary represented pairs when different domain adaptation approaches are performed on the dataset and how each pair behaves. 3k-d and 3k-n are instance adaptation approaches while 50f-3k-d-r and 50f-3k-d-b are hybrid approaches with reduced and original feature sets, respectively while retaining duplicates. The error bars show the confidence interval of each model's accuracy.*



Comparing Tables 6.1 and 6.2, we observe that the hybrid adaptation approaches used on the original numeric features perform better than the hybrid adaptation approaches used on the original binary features in six pairs, and stay within about a 3% distance for the remaining pairs. This is interesting because the corresponding models have outperformed or matched the same accuracy as in Table 6.1 despite the fact that they started with pairs that had very low accuracy to begin with in most cases.

Between the feature adaptation approach and the instance adaptation approach, which one is more effective? What parameter values result in better performance for the two approaches, respectively considering either the numerical or binary representation? As mentioned above, this question does not have a definite answer, as the Gaussian Naïve Bayes classifiers give better results for half pairs, pairs with reduced numeric features,and the Bernoulli Naïve

Bayes classifiers give better results for the other half pairs, original binary features.

Between the feature adaptation approach and the instance adaptation approach, which one is more effective? What parameter values result in better performance for the two approaches, respectively? The instance adaptation approach gives better results than the feature adaptation approach for 7 out of 12 pairs and they have a tie for 2 pairs in Table 6.1. Thus, we can say that the two approaches have complementary strengths when used with the original binary features, as the instance adaptation has performed well on pairs where feature adaptation has not performed well, and vice-versa. In Table 6.2 however, the instance adaptation performs better than the instance adaptation for all pairs.

Furthermore, we observe that feature adaptation performs better on pairs with more dissimilar source and target datasets, as opposed to the instance adaptation which performs better on pairs with more similar source and target datasets. Consequently, combining the instance based and the feature based approaches should ensure good results, as seen in our experiments. In terms of parameters, for the instance adaptation approach, the best results were obtained for $k = 3$ and $k = 5$ on binary and numeric representations, respectively. As for the number of reduced features, when comparing the hybrid models with *30f* and *50f* versus *200f*, the results are visibly better for *30f* and *50f*. The opposite is true for the feature adaptation models when considering binary representations, where better results are observed for *200f* as compared to *50f*.

When using the instance adaptation approach, is it better to keep duplicate neighbors or to remove them? When using the instance adaptation approach on the original binary representation of the data, it is better to remove duplicates regardless of the original binary or numeric representation. However, as it can be seen in Figures 6.4 and 6.5 in cases of the hybrid approach, using the reduced representation, small neighbor numbers might benefit from duplication to grant weights to the small number of source subset selected by instance adaptation. Similarly, when using the instance adaptation approach in combination with the feature adaptation approach on the original binary representation, the results are better when removing duplicates. However, the option where duplicates are retained is more beneficial when using the reduced representation with Gaussian Naïve Bayes.

Figure 6.4: *Relative frequency of having the best performance on binary representations when considering to retain or remove duplicates after instance adaptation. This frequency is calculated while having instance adaptation results from with k from 1-11 and comparing them with Gaussian Naïve Bays or Bernoulli Naïve Bays on reduced and binary subset, respectively. The error bars show the confidence interval of each model's results.*

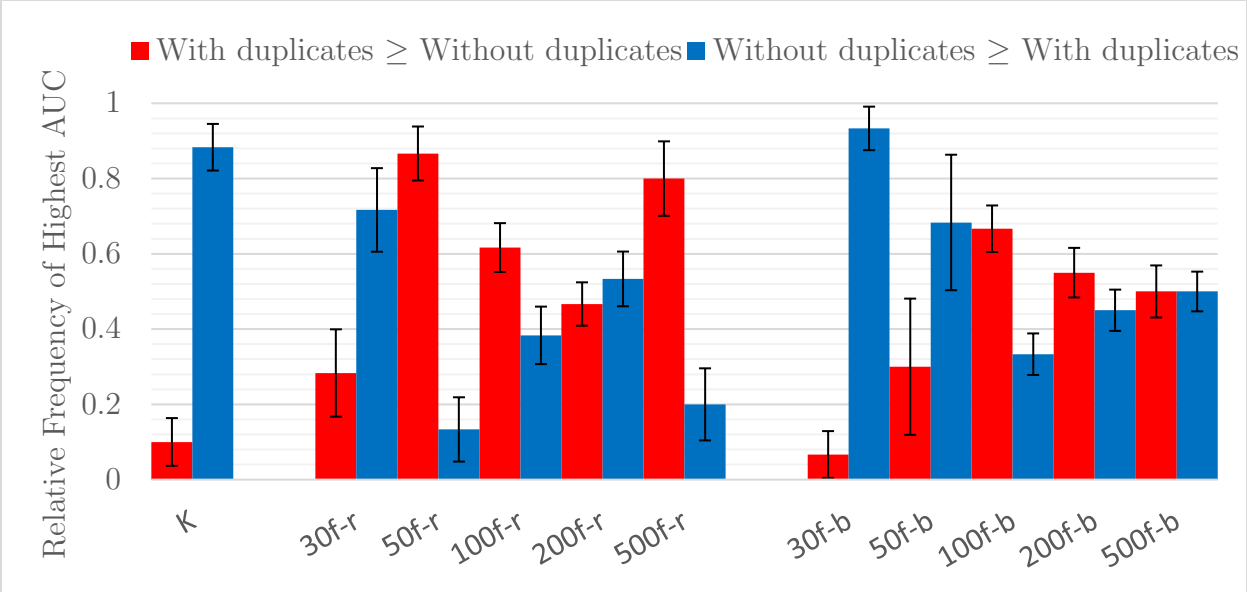


Figure 6.5: *Relative frequency of having the best performance on binary representations when considering to retain or remove duplicates after instance adaptation. This frequency is calculated while having instance adaptation results from with k from 1-11 and comparing them with Gaussian Naïve Bays or Bernoulli Naïve Bays on reduced and binary subset, respectively. The error bars show the confidence interval of each model's results.*

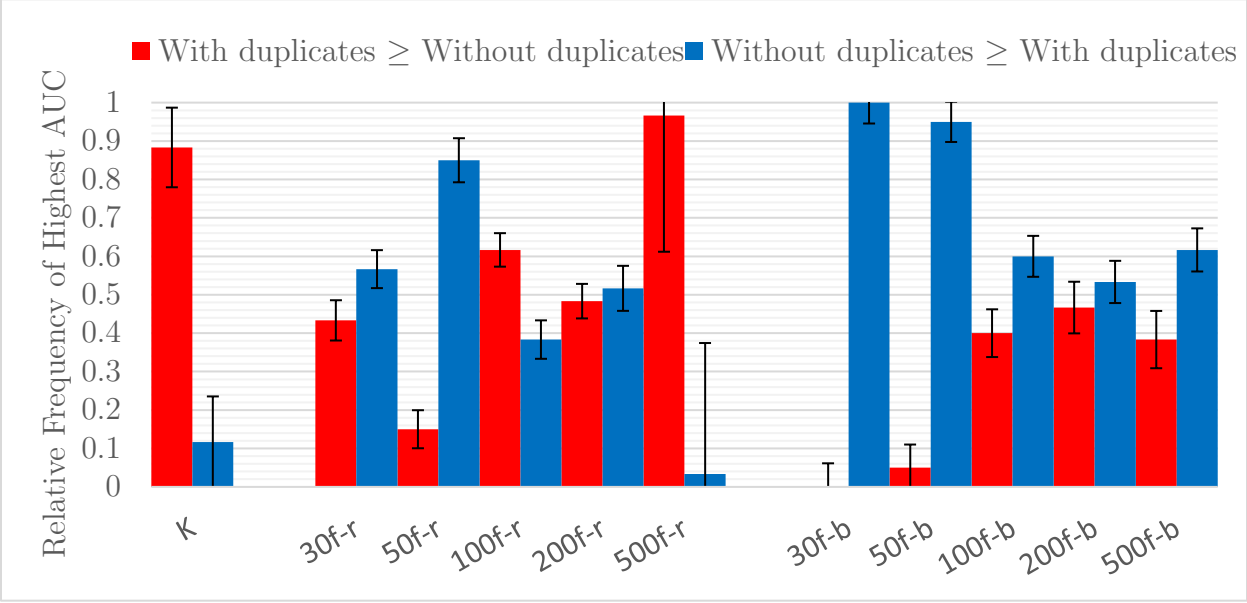


Table 6.3: *Instance-based adaptation using kNN on the binary representation, followed by Self-Training Domain Adaptation (denoted by a) and Bernoulli Naïve Bays, on the selected source subset. Accuracy results on 12 source-target pairs are shown for three values of k, specifically, 1, 3, 5 and 7, and two instance-selection settings, specifically, with duplicates (denoted by d) and without duplicates (denoted by n). For example, 3k-d means that 3 nearest neighbors are selected for each target instance, and duplicates are retained, while 3k-n means that 3 nearest neighbors are selected, but duplicates are removed. The Original results are obtained when Self-Training Domain Adaptation is used on the original binary data. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.738 ±0.012	0.843 ±0.013	0.948 ±0.003	0.872 ±0.008	0.789 ±0.007	0.750 ±0.013	0.841 ±0.004	0.711 ±0.009	0.687 ±0.009	0.808 ±0.009	0.768 ±0.010	0.772 ±0.013
Original-a	**0.862** ±0.012	**0.876** ±0.012	**0.956** ±0.003	0.855 ±0.002	0.856 ±0.008	**0.834** ±0.009	**0.872** ±0.009	0.856 ±0.008	**0.855** ±0.011	**0.885** ±0.005	**0.880** ±0.010	**0.946** ±0.004
1k-d	0.748 ±0.018	0.839 ±0.013	0.943 ±0.006	**0.878** ±0.006	0.818 ±0.006	0.746 ±0.011	0.850 ±0.006	0.753 ±0.008	0.747 ±0.011	0.831 ±0.010	0.827 ±0.005	0.783 ±0.020
1k-d-a	0.847 ±0.012	**0.870** ±0.010	0.949 ±0.004	0.859 ±0.004	**0.866** ±0.005	**0.831** ±0.010	0.869 ±0.009	0.861 ±0.006	0.832 ±0.012	0.875 ±0.005	**0.889** ±0.006	0.941 ±0.005
1k-n	0.752 ±0.014	0.837 ±0.012	0.944 ±0.003	**0.878** ±0.007	0.812 ±0.007	0.786 ±0.015	0.856 ±0.010	0.723 ±0.005	0.766 ±0.008	0.831 ±0.007	0.797 ±0.006	0.855 ±0.005
1k-n-a	0.853 ±0.014	**0.867** ±0.011	0.951 ±0.005	0.859 ±0.005	**0.866** ±0.006	**0.838** ±0.009	**0.873** ±0.011	0.848 ±0.010	**0.843** ±0.010	0.877 ±0.005	**0.879** ±0.006	**0.943** ±0.005
3k-d	0.744 ±0.014	0.842 ±0.010	0.946 ±0.005	0.871 ±0.005	0.813 ±0.005	0.728 ±0.009	0.848 ±0.008	0.759 ±0.005	0.755 ±0.007	0.832 ±0.008	0.823 ±0.006	0.820 ±0.019
3k-d-a	0.847 ±0.014	0.870 ±0.008	**0.951** ±0.004	0.852 ±0.004	0.862 ±0.004	**0.833** ±0.009	**0.868** ±0.009	**0.867** ±0.007	**0.843** ±0.007	0.878 ±0.005	**0.886** ±0.005	0.942 ±0.006
3k-n	0.752 ±0.015	0.842 ±0.010	0.946 ±0.004	0.874 ±0.005	0.806 ±0.005	0.780 ±0.006	0.853 ±0.010	0.726 ±0.009	0.747 ±0.014	0.829 ±0.008	0.789 ±0.010	0.846 ±0.011
3k-n-a	0.853 ±0.011	**0.888** ±0.040	**0.933** ±0.045	0.857 ±0.001	**0.858** ±0.012	**0.846** ±0.013	**0.867** ±0.015	0.853 ±0.010	**0.865** ±0.019	**0.886** ±0.004	**0.903** ±0.040	**0.942** ±0.005
5k-d	0.749 ±0.010	0.846 ±0.009	0.946 ±0.005	0.874 ±0.006	0.804 ±0.005	0.770 ±0.007	0.850 ±0.006	0.724 ±0.010	0.739 ±0.009	0.823 ±0.006	0.786 ±0.010	0.833 ±0.008
5k-d-a	**0.855** ±0.015	**0.870** ±0.008	0.952 ±0.005	0.849 ±0.004	0.859 ±0.004	**0.830** ±0.007	**0.871** ±0.008	**0.871** ±0.011	**0.843** ±0.005	0.880 ±0.006	**0.885** ±0.003	0.941 ±0.005
5k-n	0.736 ±0.021	0.847 ±0.010	0.949 ±0.002	0.868 ±0.007	0.811 ±0.005	0.717 ±0.005	0.848 ±0.008	0.756 ±0.008	0.758 ±0.013	0.830 ±0.007	0.819 ±0.014	0.809 ±0.008
5k-n-a	0.852 ±0.012	**0.893** ±0.039	**0.934** ±0.043	0.857 ±0.003	**0.854** ±0.013	**0.843** ±0.011	**0.869** ±0.012	0.857 ±0.006	**0.857** ±0.020	**0.883** ±0.004	**0.894** ±0.027	**0.945** ±0.004
7k-d	0.732 ±0.008	0.844 ±0.008	0.948 ±0.004	0.869 ±0.006	0.810 ±0.004	0.712 ±0.006	0.850 ±0.005	0.757 ±0.007	0.753 ±0.006	0.828 ±0.008	0.816 ±0.012	0.830 ±0.006
7k-d-a	0.856 ±0.017	**0.870** ±0.008	0.952 ±0.005	0.846 ±0.004	0.858 ±0.005	**0.829** ±0.008	**0.870** ±0.007	**0.869** ±0.009	**0.843** ±0.005	**0.880** ±0.007	**0.885** ±0.002	0.941 ±0.006
7k-n	0.746 ±0.017	0.848 ±0.009	0.947 ±0.003	**0.874** ±0.007	0.800 ±0.005	0.772 ±0.008	0.849 ±0.008	0.720 ±0.005	0.733 ±0.021	0.817 ±0.009	0.782 ±0.011	0.820 ±0.012
7k-n-a	0.850 ±0.013	**0.877** ±0.008	**0.955** ±0.004	0.856 ±0.001	**0.860** ±0.006	**0.839** ±0.007	**0.871** ±0.010	0.854 ±0.006	**0.849** ±0.009	**0.884** ±0.004	**0.880** ±0.009	0.944 ±0.004

Table 6.4: *Summary of the results for 12 source-target binary represented pairs using Self-Training Domain Adaptation (denoted by a) and Bernoulli Naïve Bays. The upper section of the table contains the individual instance adaptation (1k-d and 1k-n) approaches, while the bottom section contains the hybrid approach for 30f through 200f , and 1k. The standard deviation of cross-validation is shown under each result (denoted by ±). Significant best results for each pair are highlighted in bold (based on a t-test with $p \leq 0.05$).*

Source Target	BB AF	BB OT	BB WT	OT AF	QF AF	QF BB	QF OT	SH AF	SH BB	SH OT	SH QF	SH WT
Original	0.738 ±0.012	0.843 ±0.013	0.948 ±0.003	0.872 ±0.008	0.789 ±0.007	0.750 ±0.013	0.841 ±0.004	0.711 ±0.009	0.687 ±0.009	0.808 ±0.009	0.768 ±0.010	0.772 ±0.013
Original-a	**0.862** ±0.012	**0.876** ±0.012	**0.956** ±0.003	0.856 ±0.008	0.856 ±0.008	**0.834** ±0.009	**0.872** ±0.009	0.856 ±0.008	**0.855** ±0.011	0.875 ±0.005	**0.889** ±0.006	**0.946** ±0.004
1k-d	0.748 ±0.018	0.839 ±0.013	0.943 ±0.006	**0.878** ±0.006	0.818 ±0.006	0.746 ±0.011	0.850 ±0.006	0.753 ±0.008	0.747 ±0.011	0.831 ±0.010	0.827 ±0.005	0.783 ±0.020
1k-d-a	0.849 ±0.013	**0.870** ±0.010	0.949 ±0.004	0.866 ±0.005	0.866 ±0.005	**0.831** ±0.010	0.869 ±0.009	**0.861** ±0.006	0.832 ±0.012	0.877 ±0.005	0.879 ±0.006	0.941 ±0.005
1k-n	0.752 ±0.014	0.837 ±0.012	0.944 ±0.003	**0.878** ±0.007	0.812 ±0.007	0.786 ±0.015	0.856 ±0.010	0.723 ±0.005	0.766 ±0.008	0.831 ±0.007	0.797 ±0.006	0.855 ±0.005
1k-n-a	0.847 ±0.012	0.867 ±0.011	0.951 ±0.005	0.866 ±0.006	0.866 ±0.006	0.838 ±0.009	**0.873** ±0.011	0.848 ±0.010	0.843 ±0.010	0.864 ±0.005	0.879 ±0.009	**0.943** ±0.005
30f-1k-d-b	0.788 ±0.003	0.836 ±0.009	0.920 ±0.008	**0.878** ±0.006	0.765 ±0.008	0.713 ±0.014	0.827 ±0.006	0.800 ±0.013	0.740 ±0.009	0.826 ±0.006	0.862 ±0.008	0.846 ±0.005
30f-1k-d-b-a	**0.868** ±0.016	**0.874** ±0.006	0.947 ±0.007	**0.870** ±0.011	**0.875** ±0.009	**0.836** ±0.009	0.871 ±0.010	**0.867** ±0.009	0.839 ±0.008	0.873 ±0.006	**0.891** ±0.009	0.939 ±0.003
30f-1k-n-b	0.782 ±0.008	0.833 ±0.007	0.936 ±0.008	**0.878** ±0.006	0.798 ±0.009	0.769 ±0.009	0.840 ±0.009	0.796 ±0.012	0.777 ±0.008	0.847 ±0.006	0.862 ±0.009	0.897 ±0.007
30f-1k-n-b-a	0.862 ±0.017	**0.873** ±0.006	0.951 ±0.003	0.875 ±0.009	**0.838** ±0.007	**0.841** ±0.006	**0.874** ±0.012	0.861 ±0.011	**0.847** ±0.008	0.866 ±0.007	**0.889** ±0.024	0.943 ±0.004
50f-1k-d-b	0.780 ±0.009	0.837 ±0.012	0.923 ±0.010	**0.878** ±0.006	0.788 ±0.009	0.727 ±0.011	0.835 ±0.006	0.793 ±0.010	0.771 ±0.007	0.823 ±0.005	0.859 ±0.003	0.872 ±0.009
50f-1k-d-b-a	0.858 ±0.014	0.868 ±0.007	0.945 ±0.005	**0.868** ±0.007	0.868 ±0.007	0.828 ±0.008	**0.866** ±0.005	0.861 ±0.007	0.841 ±0.009	0.871 ±0.006	0.881 ±0.007	0.937 ±0.004
50f-1k-n-b	0.771 ±0.010	0.833 ±0.012	0.938 ±0.004	**0.878** ±0.006	0.804 ±0.006	0.770 ±0.005	0.841 ±0.007	0.788 ±0.010	0.783 ±0.012	0.846 ±0.010	0.850 ±0.005	0.899 ±0.009
50f-1k-n-b-a	0.846 ±0.014	0.868 ±0.008	0.950 ±0.006	**0.868** ±0.007	0.870 ±0.011	0.836 ±0.007	**0.871** ±0.010	0.856 ±0.010	**0.844** ±0.009	0.868 ±0.004	0.884 ±0.008	0.942 ±0.005
100f-1k-d-b	0.778 ±0.012	0.843 ±0.010	0.938 ±0.007	**0.878** ±0.006	0.815 ±0.006	0.729 ±0.013	0.843 ±0.003	0.809 ±0.012	0.726 ±0.013	0.816 ±0.005	0.853 ±0.006	0.898 ±0.006
100f-1k-d-b-a	0.851 ±0.016	0.865 ±0.010	0.944 ±0.005	0.851 ±0.003	0.857 ±0.006	0.824 ±0.006	0.862 ±0.006	0.857 ±0.005	0.834 ±0.007	0.873 ±0.006	**0.882** ±0.007	0.933 ±0.003
100f-1k-n-b	0.770 ±0.014	0.843 ±0.011	0.943 ±0.007	**0.878** ±0.006	0.816 ±0.009	0.774 ±0.008	0.853 ±0.006	0.787 ±0.016	0.747 ±0.025	0.839 ±0.009	0.850 ±0.007	0.903 ±0.003
100f-1k-n-b-a	0.839 ±0.015	0.863 ±0.010	0.949 ±0.004	0.857 ±0.006	**0.864** ±0.009	0.824 ±0.005	0.864 ±0.009	0.852 ±0.006	0.835 ±0.012	**0.885** ±0.005	**0.880** ±0.010	0.941 ±0.002
200f-1k-d-b	0.769 ±0.012	0.851 ±0.011	0.941 ±0.004	**0.878** ±0.006	0.824 ±0.005	0.725 ±0.016	0.858 ±0.010	0.808 ±0.013	0.786 ±0.013	0.826 ±0.008	0.859 ±0.005	0.908 ±0.005
200f-1k-d-b-a	0.842 ±0.010	0.862 ±0.011	0.939 ±0.006	0.838 ±0.007	0.851 ±0.004	**0.824** ±0.021	0.858 ±0.010	0.842 ±0.008	0.832 ±0.016	0.872 ±0.007	0.877 ±0.004	0.935 ±0.003
200f-1k-n-b	0.768 ±0.011	0.853 ±0.010	0.946 ±0.003	**0.878** ±0.006	0.825 ±0.008	0.771 ±0.011	0.858 ±0.011	0.795 ±0.016	0.790 ±0.019	0.848 ±0.009	0.862 ±0.007	0.906 ±0.004
200f-1k-n-b-a	0.833 ±0.009	0.860 ±0.012	0.947 ±0.005	0.851 ±0.004	0.851 ±0.003	0.818 ±0.010	0.861 ±0.009	0.838 ±0.010	0.820 ±0.010	0.867 ±0.004	**0.883** ±0.007	0.939 ±0.004

How do the feature and hybrid approaches affect the Self-Training Domain Adaptation classifier? Table 6.3 shows that feature adaptation outperforms or matches the original binary results for all the pairs proving feature adaptation to be facilitative to domain adaptation classification. In Table 6.4, the hybrid approach outperforms or matches the best in 11 cases and have very close values in the other pair. Both *30f-1k-d-b-a* and *30f-1k-n-b-a* perform very well which makes *30f* the best feature adaptation and *1k* the best instance adaptation approach.

Chapter 7

Related Works and Conclusion

7.1 Related Works

Machine learning algorithms have been used to help responders sift through the huge amount of crisis data, and prioritize information that may be useful for response and relief (Verma et al., 2011; Caragea et al., 2011; Vieweg, 2012; Terpstra et al., 2012; Purohit et al., 2013; Imran et al., 2013; Caragea et al., 2014; Ashktorab et al., 2014; Sen et al., 2015; Huang and Xiao, 2015; Imran et al., 2016a). For example, Imran et al. (2013) used conditional random fields to find tweets within specific situational awareness categories. Sen et al. (2015) used Support Vector Machine (SVM) classifiers to differentiate between situational and non-situational tweets. Huang and Xiao (2015) introduced a detailed list of situational awareness categories, divided based on three stages of a disaster (preparedness, emergency response, and recovery), and used k-Nearest Neighbors, Logistic Regression and Naïve Bayes classifiers to automatically classify tweets with respect to the categories defined.

While research on supervised machine learning in the area of emergency response has shown that it is possible to automatically classify disaster-related data, it has also emphasized one of the most important challenges that precludes the use of supervised machine learning in real time in an emerging crisis situation: *the lack of labeled data to train reliable supervised models as the crisis unfolds.* To address this challenge, several works proposed to use labeled

data from prior "source" crises to learn *supervised* classifiers for a "target" crisis (Verma et al., 2011; Imran et al., 2016b; Caragea et al., 2016; Nguyen et al., 2017). One drawback of this approach is that supervised classifiers learned in one crisis event, do not generalize well to other events (Qadir et al., 2016; Imran et al., 2015), as each event has unique characteristics (Palen and Anderson, 2016). Domain adaptation approaches (Pan and Yang, 2010; Jiang, 2008) that make use of unlabeled data from the target disaster in addition to label data from a source disaster are desirable. Some recent works (Li et al., 2015, 2017a,b) have shown the using domain adaptation approaches can significantly improve the results of the supervised classifiers learned from source only. According to Pan and Yang (2010), domain adaptation is achieved by performing parameter adaptation, feature adaptation or instance adaptation. A comprehensive description of works in each category can be found in (Pan and Yang, 2010).

In the space of disasters, the domain adaptation approaches proposed by Li et al. (2015; 2017b) can be seen as parameter-based adaptation approaches. To the best of our knowledge, there are no instance-based or feature-based adaptation approaches that have been used for classifying disaster related data. As a consequence, in this study we focus specifically on a hybrid approach that combines feature-based adaptation based on matrix factorization with instance-based adaptation based on the kNN algorithm, and compare the hybrid approach with the individual feature-based and instance-based approaches.

7.2 Conclusions and Future Work

Social media data taken from sources such as Twitter contain invaluable data which can be used in times of crisis and emergency situations to improve response and awareness. Despite many supervised learning approaches being proposed, not many agencies and groups use these approaches to identify useful information, due to lack of labeled data for training the supervised models. In this study, we proposed a simple but powerful feature-instance adaptation approach to reduce the variation between source and target disasters. Combined with Naïve Bayes classifiers, the proposed adaptation approach produces accuracy results

that are significantly better than the results of the supervised models learned from source alone, in some cases by more than 12%, when used for the task of identifying tweets related to a particular disaster.

The CrisisLexT6 dataset was used to construct twelve pairs of disasters that we experimented with. Our results showed that adaptation-based models perform significantly better than the supervised models. We also showed that feature adaptation and instance adaptation approaches have complementary strengths that can be combined to produce better results. We further revealed that when using instance adaptation, the original binary representation performs best when duplicates are retained, on the other hand when feature and instance adaptations are combined we achieve the best results when duplicates are removed. We argued that the hybrid feature-instance adaptation approaches are more reliable due to their consistent competitive results, especially when not considering duplicates for the instance adaptation step. Overall, the results of this study can be used to recommend the best options and parameters for the adaptation approaches, based on our observations on 12 different pairs of disasters.

Performing classification on the numeric as well as binary representation, we have argued that we can attain the best results when our hybrid approach is used to transform the numeric representation and the instances are selected from the binary representation. This combines the strengths of specificity of the numeric representations and the accuracy of classification of the binary representation. We also showed that using the feature-instance hybrid approach can not only improve accuracy in simpler models such as Naïve Bays but also more complex domain adaptation models such as Self-Training Domain Adaptation.

In future work, more experiments can be done using different classifiers, including deep learning classifiers, on the selected source data. Furthermore, different matrix factorization and clustering approaches (potentially, with different distance metrics) can be explored.

Bibliography

Ashktorab, Z., Brown, C., Nandi, M., and Culotta, A. (2014). Tweedr: Mining twitter to inform disaster response. *Proc. of ISCRAM*.

Beigi, G., Hu, X., Maciejewski, R., and Liu, H. (2016). An overview of sentiment analysis in social media and its applications in disaster relief. In *Sentiment Analysis and Ontology Engineering*, pages 313–340. Springer.

Bullock, J., Haddow, G., and Coppola, D. P. (2012). *Homeland security: the essentials*. Butterworth-Heinemann.

Caragea, C., McNeese, N., Jaiswal, A., Traylor, G., Kim, H.-W., Mitra, P., Wu, D., Tapia, A. H., Giles, C. L., Jansen, B. J., and Yen, J. (2011). Classifying text messages for the haiti earthquake. In *Proceedings of the 8th International Conference on Information Systems for Crisis Response and Management*, ISCRAM '11, Lisbon, Portugal.

Caragea, C., Silvescu, A., and Tapia, A. H. (2016). Identifying informative messages in disasters using convolutional neural networks. In *13th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Rio de Janeiro, Brasil, May 22-25, 2016.*

Caragea, C., Squicciarini, A. C., Stehle, S., Neppalli, K., and Tapia, A. H. (2014). Mapping moods: Geo-mapped sentiment analysis during hurricane sandy. In *11th Proceedings of the International Conference on Information Systems for Crisis Response and Management, University Park, Pennsylvania, USA, May 18-21, 2014.*

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Guo, W. and Diab, M. (2012). A simple unsupervised latent semantics based approach for sentence similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 586–590. Association for Computational Linguistics.

Huang, Q. and Xiao, Y. (2015). Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery. *ISPRS International Journal of Geo-Information*, 4(3):1549–1568.

Imran, M., Castillo, C., Diaz, F., and Vieweg, S. (2015). Processing social media messages in mass emergency: A survey. *ACM Computing Surveys (CSUR)*, 47(4):67.

Imran, M., Chawla, S., and Castillo, C. (2016a). A robust framework for classifying evolving document streams in an expert-machine-crowd setting. In *Proceedings of the 18th International Conference on Data Mining (ICDM)*, Barcelona, Spain.

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., and Meier, P. (2013). Practical extraction of disaster-relevant information from social media. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pages 1021–1024.

Imran, M., Mitra, P., and Srivastava, J. (2016b). Cross-language domain adaptation for classifying crisis-related short messages. In *13th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Rio de Janeiro, Brasil, May 22-25, 2016.*

Jiang, J. (2008). A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 3.

Li, H., Caragea, D., and Caragea, C. (2017a). Towards practical usage of a domain adaptation algorithm in the early hours of a disaster. In *Proceedings of the 14th International Con-*

ference on Information Systems for Crisis Response and Management (ISCRAM 2017), France.

Li, H., Guevara, N., Herndon, N., Caragea, D., Neppalli, K., Caragea, C., Squicciarini, A., and Tapia, A. (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, Kristiansand, Norway*.

Li, H., Herndon, N., Caragea, D., and Caragea, C. (2017b). Disaster response aided by tweet classification with a domain adaptation approach. *Journal of Contingencies and Crisis Management (JCCM), Special Issue on HCI in Critical Systems. In press*.

Lin, C.-J. (2007). Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779.

Meier, P. (2015). *Digital Humanitarians: How Big Data Is Changing the Face of Humanitarian Response*. CRC Press, Inc., Boca Raton, FL, USA.

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

Nguyen, D. T., Al-Mannai, K., Joty, S. R., Sajjad, H., Imran, M., and Mitra, P. (2017). Robust classification of crisis-related data on social networks using convolutional neural networks.

Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. (2014). Crisislex: A lexicon for collecting and filtering microblogged communications in crises.

Palen, L. and Anderson, K. M. (2016). Crisis informatics-new data for extraordinary times. *Science*, 353(6296):224–225.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359.

Purohit, H., Castillo, C., Diaz, F., Sheth, A., and Meier, P. (2013). Emergency-relief coordination on social media: Automatically matching resource requests and offers. *First Monday*, 19(1).

Qadir, J., Ali, A., Rasool, R. U., Zwitter, A., Sathiaseelan, A., and Crowcroft, J. (2016). Crisis analytics: Big data driven crisis response. *CoRR*, abs/1602.07813.

Sen, A., Rudra, K., and Ghosh, S. (2015). Extracting situational awareness from microblogs during disaster events. In *Communication Systems and Networks (COMSNETS), 2015 7th International Conference on*, pages 1–6. IEEE.

Terpstra, T., De Vries, A., Stronkman, R., and Paradies, G. (2012). *Towards a realtime Twitter analysis during crises for operational crisis management*. Simon Fraser University.

Verma, S., Vieweg, S., Corvey, W. J., Palen, L., Martin, J. H., Palmer, M., Schram, A., and Anderson, K. M. (2011). Natural language processing to the rescue? extracting "situational awareness" tweets during mass emergency. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*.

Vieweg, S. E. (2012). *Situational awareness in mass emergency: A behavioral and linguistic analysis of microblogged communications*. PhD thesis.

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