MultiFOLD: Multi-source Domain Adaptation For Offensive Language Detection

Aymé Arango*[1], Chia-Jung Lee[2], Sheikh Muhammad Sarwar[3], Vanessa Murdock[2], Parisa Kaghazgaran[3]

[1] University of Chile
[2] Amazon AWS AI/ML
[3] Amazon.com
arangoayme@gmail.com, [cjlee, smsarwar, parisaka]@amazon.com, vmurdock@acm.org

Abstract

Automatic offensive language detection remains challenging, and is a crucial part of preserving the openness of digital spaces, which are an integral part of our everyday experience. The ever-growing forms of offensive online content makes traditional supervised approaches harder to scale due to the financial and psychological costs incurred by collecting human annotations. In this work, we propose a domain adaptation framework for offensive language detection, MultiFOLD, which learns and adapts from multiple existing data sets (or source domains) to an unlabeled target domain. Under the hood, a curriculum learning algorithm is employed that kicks off learning with the instances most similar to the target domain while gradually expanding to more distant instances. The proposed model is trained with a standard task-specific loss and a domain adversarial objective which aims to minimize the language distinctions across the multiple sources and the target, allowing the classifier to distinguish offensiveness rather than domain. Our experiments on six publicly available data sets demonstrate the effectiveness of MultiFOLD. Relative improvement in F1 of 0.5% (WOAH) to 29.7% (ICWSM) is found across five out of the six datasets compared to the state-of-the-art domain adaptation baseline BERT-DAA, resulting in an average of 6% relative F1-score gain.

1 Introduction

Offensive language detection is a key component of online content moderation, vital for maintaining inclusive online spaces, maintaining trust in organizations and businesses, given a global and diverse online community, with disparate expectations of what is appropriate. Offensive language encompasses a range of expression from sensitive content such as language related to sexuality, drug use, or health, to profanity, to harmful content such as hate speech and online bullying. (Davidson et al. 2017; Golbeck 2023; Arora et al. 2023). To automatically detect and filter such content, prior work has investigated supervised machine learning approaches on diverse data (Waseem and Hovy 2016; Chatzakou et al. 2017; Gambäck and Sikdar 2017).

While these supervised learning approaches achieve impressive performance in test sets that are sampled from the same domain as the training set, they fail to achieve the same performance on data from a new domain, i.e. data that targets a different identity group, represents a different topic, or contains other linguistic variants (such as slang, spelling variants, new dialects). The task inevitably becomes harder to scale due to challenge of curating high quality labeled data that covers the breadth of vocabulary associated with different offensive language domains. Not only is offensive language online ever evolving, existing data sets by construction are static slices of language from a particular place and time. They represent a relatively limited vocabulary, are potentially biased toward the authors (Arango, Pérez, and Poblete 2019), the target of the offense (such as ethnicity or age) (Davidson, Bhattacharya, and Weber 2019) or the topic (Wiegand, Ruppenhofer, and Kleinbauer 2019).

Previous approaches to addressing insufficient (or absent) in-domain annotations often rely on adapting models learned from resource-rich to resource-scarce domains. One line of research (Karan and Šnajder 2018; Caselli et al. 2020) considered adapting models from a single source domain to a target domain in a few-shot setting. While adaptation based on fine-tuning has been demonstrated to be effective, it is not always possible to get annotated data or examples, especially for sensitive personal collections such as emails or text messages. For this reason researchers have turned to unsupervised domain adaptation (Ganin and Lempitsky 2015; Bose, Illina, and Fohr 2021b; Ryu, Lee, and Lee 2022), which leverages source domain labels as well as unlabeled data from the target domain. Prior work also showed that augmenting a source domain with synthetic data can improve offensive content detection on a target domain (Sarwar and Murdock 2022).

One primary challenge with domain adaptation is the gap between the source and target domain vocabularies. Arango, Pérez, and Poblete (2019) identified that the performance of neural models trained using one data set may drop significantly when tested on another. To improve model generalization, they proposed to randomly shuffle instances from multiple source domains for training. Their approach treats the instances from multiple sources with equal importance, regardless of their similarity to the target. We hypothesize that a model should learn more from instances that are more similar to the target domain. Our analysis in Section 5.3 shows that different source models result in highly variable

*Work done while the author was an intern at Amazon.com

prediction performance, and therefore vary greatly in their utility for domain adaptation. We hypothesize that a source instance selection strategy, which takes into account the relationship between the source and the target, will significantly improve the effectiveness of the adaptation.

In this paper, we propose a **Multi**-source Domain Adaptation Framework **f**or **O**ffensive **L**anguage **D**etection, (MultiFOLD) that applies an Automatic Curriculum Learning (CL) algorithm (Bengio et al. 2009) to select source domain instances based on a dynamic threshold as well as a semantic representation, while training an offensive language classifier with task-specific and domain-adversarial objectives.

Our framework is an orchestration of two primary components, namely the *Difficulty Measurer* and the *Training Scheduler*. The Difficulty Measurer sorts training instances from multiple source domains to form a curriculum in a progressive manner. In each epoch, all source instances are sorted by their similarity to target domain data according to their textual representations learned from the previous epoch. The more similar a source instance is to the target, the easier it is considered for the purpose of adaptation. Different from prior work (Zhang et al. 2019; Guo, Pasunuru, and Bansal 2020) which sorts instances by difficulty once and for all, our CL algorithm is designed to gradually lower the similarity threshold to include more challenging source instances over time, following an easy-to-hard curriculum.

After the sorting is done, the Training Scheduler selects which source instances to use for training, according to a dynamic threshold. The model is trained on the selected instances, with a task-specific cross entropy loss, guided by the source domain labels. It employs domain adversarial training through reverse gradients. Domain adversarial training learns domain-agnostic semantic representations, such that the text distributions differ minimally across source and target domains. The learned and updated representations are consumed by the Difficulty Measurer for the next epoch of similarity computation, forming a mutually assistive cycle.

To further improve the generalization from the source to the target, we augment the dense semantic representations with the Hurtlex lexicon (Bassignana, Basile, and Patti 2018). Hurtlex is a human-curated lexicon composed of common offensive terms. We used Hurtlex to represent text instances with one-hot representations of their offensive content.

We conducted extensive experiments on six publicly available data sets with a variety of offensive language forms. Each data set is rotated and elected to be a target domain while the rest are kept as source domains.We do not consider the target domain offensiveness labels for any training purpose. We compared to multiple baselines, including the state-of-the-art BERT-DAA method (Ryu, Lee, and Lee 2022) for domain adaptation. MultiFOLD improves over the baselines across all target domains, by as much as 7% relative improvement in the F1 score. Further ablation analysis also highlights :

- The dynamic selection of instances effectively adapts from multiple source domains to the target compared to a predefined curriculum
- Incorporating Hurtlex as part of the representation abstracts the forms of offensive languages and significantly improves results
- Learning generic semantic representations via the adversarial objective further minimizes the linguistic gap across domains

Overall, the results demonstrate the effectiveness of domain adaptation under MultiFOLD.

2 Related Work

We discuss related work in three areas that are closely related to offensive language detection through cross-domain, multi-source domain, and curriculum learning techniques for domain adaptation. A summary of the related work can be found at Table 1. It illustrates that our proposed framework covers different aspects of domain adaption.

2.1 Cross-Domain Adaptation

Cross-domain adaptation transfers information from a **single** source domain into a target domain in settings where either no labeled examples (zero-shot) or a few labeled examples (few-shot) from the target are available. The simplest domain adaptation approach is to fine-tune a generic model using samples from a specific domain (Karan and Šnajder 2018; Caselli et al. 2020). This approach is limited by the availability of labeled data in the target domain.

For the task of offensive content detection, prior work addressed de-biasing (Arango, Pérez, and Poblete 2019), data augmentation (Sarwar and Murdock 2022) and generalization (Ludwig et al. 2022; Bose, Illina, and Fohr 2021a; Pamungkas, Basile, and Patti 2021; Pamungkas and Patti 2019; Wiegand, Ruppenhofer, and Kleinbauer 2019; Koufakou et al. 2020; Sarwar et al. 2022). Owing to a lack of labeled data for the target domain, Sarwar and Murdock (2022) proposed to generate synthetic training examples by inserting target domain hate terms into generic templates. In contrast, our work addresses data scarcity by using real examples, but from multiple different sources.

To remedy low performance on the target domain, several recent works consider generalization techniques to align source and target domains. In one direction, several works (c.f. Ludwig et al. (2022); Bose, Illina, and Fohr (2021a,b); Ryu, Lee, and Lee (2022)) introduce an adversarial loss in addition to classification loss to generalize language-specific attributes between source and target .

In another direction, given that the definition of what is considered offensive can vary across different platforms and types of hateful speech, several approaches used the Hurtlex vectors as an abstraction of offensiveness (Pamungkas, Basile, and Patti 2021; Pamungkas and Patti 2019; Wiegand, Ruppenhofer, and Kleinbauer 2019; Koufakou et al. 2020) or penalized hate terms existing in the source domain to reduce their importance (Bose et al. 2022a,b).They hypothesize that domain-specific terms restrict the classifier's ability to adapt to a new domain with different types of hate speech, so they propose giving more attention to contextual words instead.

	Multi-source	Zero-shot	Linguistic generalization	Offensiveness abstraction	Adaptive strategy
(Sarwar and Murdock 2022)	×	✓	×	✓	×
(Bose, Illina, and Fohr 2021b)	×	✓	✓	×	×
(Bose, Illina, and Fohr 2021a)	×	×	✓	×	×
(Bose et al. 2022a)	×	✓	×	✓	×
(Bose et al. 2022b)	×	×	×	✓	×
(Plaza-Del-Arco et al. 2021)	✓	×	×	×	×
(Arango, Pérez, and Poblete 2019)	✓	✓	×	×	×
MultiFOLD	✓	✓	✓	✓	✓

Table 1: Literature comparison of cross-domain offensive language detection

Our model is able to adapt to both linguistic and offensiveness generalizations by employing adversarial loss and Hurtlex vectors respectively.

Several works consider cross domain adaptation for other tasks such as fake news detection (Mosallanezhad et al. 2022), machine translation (Zhang et al. 2019), sentiment classification (Guo, Pasunuru, and Bansal 2020; Liu et al. 2021; Sun, Feng, and Saenko 2016), cross-lingual embedding representation (Aluru et al. 2020) and object recognition (Sun, Feng, and Saenko 2016) but these approaches are not suitable for offensive language detection.

Our work differs from previous research in that it (a) leverages labeled data from multiple sources, (b) is able to generalize both linguistically and for offensiveness, and (c) can be adapted to a new domain in a zero-shot setting.

2.2 Multi-Source Domain Adaptation

Multi-source domain adaptation seeks to leverage data from multiple sources to improve offensive content detection in a target domain (Corazza et al. 2019; Pamungkas, Basile, and Patti 2020). Plaza-Del-Arco et al. (2021) augments hate speech training data with sentiment data. They do not consider any particular strategy to combine different data sources and mainly transfer knowledge to the target domain in a supervised or few-shot setting by using labeled samples from the target domain.

In contrast, our work (a) employs an automatic Curriculum Learning (CL) strategy to effectively combine multiple sources; and (b) takes a zero-shot approach without any additional data from the target domain.

2.3 Curriculum Learning

Curriculum Learning (CL) was first introduced in Bengio et al. (2009), who showed that models can generalize effectively if the training examples are not randomly presented, but organized meaningfully. Intuitively, the learning process of neural networks, emulating humans, attempts to learn less complex examples first. Hence, CL initiates model training with the easiest examples and increases the complexity of the training examples with each training epoch. The difficulty of the examples is computed as their embedding distance to the target domain. In a typical CL setting, data points are organized *a priori* in pre-defined batches sorted by difficulty. As the training progresses, the more difficult batches become available until the full data set has been used (Zhang et al. 2019; Guo, Pasunuru, and Bansal 2020). In contrast to prior work, our proposed CL technique dynamically selects samples during the training process. It learns the semantic representation of each instance, leading to evolving similarities between target and source samples, rather than a constant similarity estimated one time prior to training. This approach provides a dynamic representation of the samples as training progresses.

Several works consider CL for domain adaptation in different tasks such as sentiment analysis (Rao 2020; Zhao et al. 2021; Guo, Pasunuru, and Bansal 2020), machine translation (Zhang et al. 2019; Zhan et al. 2021), data augmentation (Ludwig et al. 2022) and object detection (Yang et al. 2020; Soviany et al. 2022).

To the best of our knowledge, ours is the first work to explore CL as an effective sample selection strategy in offensive language detection. Our work leverages CL to effectively combine different sources, selects samples dynamically and demonstrates strong performance in the absence of any labeled data from the target domain.

3 Problem Definition

We address the multi-source, unlabeled domain adaptation problem for offensive textual content detection. Precisely, we have a collection, $S = \{S_1, S_2, \dots S_n\}$, of n source datasets from different domains, and a target-domain unlabeled dataset T. Each source dataset S_l is labeled, *i.e.,* $S_l = \left\{(x_i^{S_l}, y_i^{S_l})\right\}_{i=1}^{|S_l|}$, where $x_i^{S_l}$ is the textual content, $y_i^{S_l} \in \{\textit{offensive}, \textit{neutral}\}$ is the label of $x_i^{S_l}$, and $\mid S_l \mid$ is the number of labeled examples in S_l. As we focus on detecting offensive content, we set the label of any non-offensive content as *neutral*. In our target-domain data set, $T = \left\{(x_j^T)\right\}_{j=1}^{|T|}$, we have $\mid T \mid$ textual examples without any labels. Our goal is to design a *training strategy* that leverages S and T to learn a model M, such that the resulting M can be effectively applied to identifying offensive content in the target domain. We want M to learn the characteristics of offensive language from S, and the characteristics of the target domain from T, so that we can successfully apply M to the target data set.

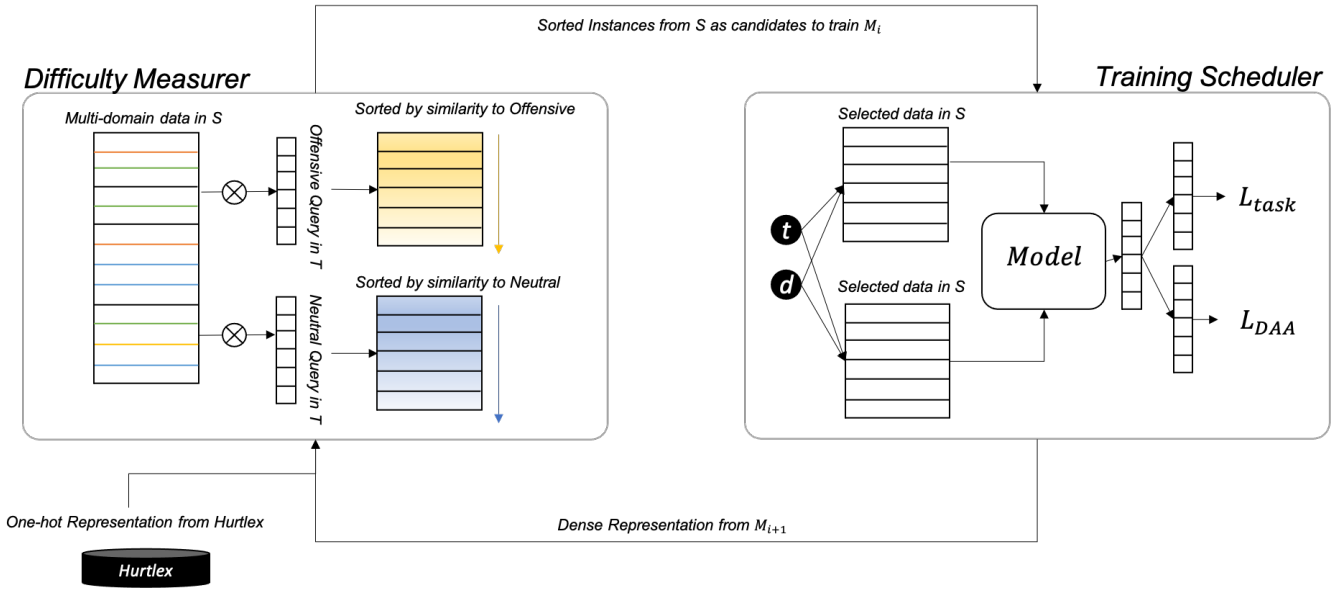

Figure 1: Proposed MultiFOLD Framework composed of two main components: Difficulty Measurer (orders training instances based on their similarity to the target domain and it is enhanced by offensiveness semantic space a.k.a Hurtlex lexicon vectors) and Training Scheduler (selects a subset of the ordered instances for training based on a threshold t and a decay factor d which relaxes the threshold for sampling instances in each iteration). This component is enhanced by adversarial domain adaptation loss a.k.a L_{ADA}) in addition to the classification loss a.k.a L_{task}.

4 Methodology

As we want to optimize the performance of M on the target domain, we hypothesize that we create a relatively easier task for M by asking it to optimize on instances from the source datasets that are similar to the target dataset. Our hypothesis is based on Curriculum Learning (CL), where it is assumed that a model learns better when it is trained on progressively harder examples (Wang, Chen, and Zhu 2022). We adapt this notion and train M starting with source-domain instances that are semantically similar to the target domain and gradually add harder instances that are more distant from the target domain. As our application area is offensive language detection, we propose to compute the similarities in both generic semantic space and a semantic space for offensive language.

MultiFOLD has two components : a *Difficulty Measurer* and a *Training Scheduler*. Before each training epoch, the difficulty measurer ranks the training instances in increasing order of difficulty, i.e. it ranks the training instances in S based on their similarity to the target domain. In a typical CL setting, the difficulty measurer estimates the difficulty of the instances before the training process begins. Based on the intuition that it is important to re-estimate the difficulty of the examples as the training progresses, the similarities of the instances in S with respect to T are updated at each epoch because the similarity calculation is based on the model M, which updates in each epoch. This makes our CL approach dynamic rather than pre-defined. In addition to the difficulty measurer, we propose a training scheduler that selects instances from similarity-ranked S based on a *similarity threshold* and a *decay factor*.

More formally, in MultiFOLD at each epoch e, a model M optimizes its loss for a *subset* of labeled instances from S that are *similar* to T. We use the model checkpoint at the previous epoch, $e-1$ to obtain the representations for similarity calculation. We provide a diagram of our proposed framework in Figure 1.

In the following sections, we describe MultiFOLD component by component, starting with the model M which is based on a pre-trained transformer with an Adversarial Domain Adaptation (ADA) component. The ADA component ensures that the representation of M captures domain-agnostic features from the source and target domains for offensive language detection.

4.1 Adversarial Domain Adaptation Classifier

As mentioned above, the model M is classifier with a transformer-based representation architecture and an Adversarial Domain Adaptation (ADA) component. The representation learning component R, of the model M, is a pre-trained transformer. Given a sequence of l tokens, $x_i = (x_{i1}, x_{i2}, \ldots x_{il})$, R generates contextualized embeddings of each of the tokens in the sequence, $\vec{x_i} = (\vec{x_{i1}}, \vec{x_{i2}}, \ldots, \vec{x_{il}})$, using multi-head attention. Following existing work on text classification, we further add a Bi-directional Long Short-term Memory (Bi-LSTM) layer that makes forward and backward passes over the token embedding sequence $\vec{x_i}$, computed by R, and generates h_f, and h_b, respectively. We concatenate h_f and h_b and apply a fully-connected linear classification layer C generating two logit outputs. We summarize this process in the following equations:

$$\vec{x_{i1}}, \vec{x_{i2}}, \ldots, \vec{x_{il}} = R(x_{i1}, x_{i2}, \ldots x_{il}) \quad (1)$$

$$h_f, h_b = LSTM(\vec{x_{i1}}, \vec{x_{i2}}, \ldots, \vec{x_{il}}) \quad (2)$$

We optimize the logit outputs of the classifier $C([h_f, h_b])$ with respect to the gold label y_i using a softmax cross-entropy loss. Our loss function is as follows:

$$\underset{R,LSTM,C}{\operatorname{argmin}} \mathcal{L}(C([h_f, h_b]), y_i) \quad (3)$$

Our model design includes a representation of x_i agnostic to any domain, using an ADA component. After obtaining the token vector representation of x_i from R, i.e., $\vec{x_i} = (\vec{x_{i1}}, \vec{x_{i2}}, \ldots, \vec{x_{il}})$, we average pool the token vectors, and then pass the pooled representation to the ADA. The ADA is a domain classifier that takes the mean-pooled representation as input and outputs the index of the data set d_i that x_i is a member of. Note that we use a number of source domain data sets for training, and for each data point, we have the ground truth data set label. Moreover, we use the target-domain unlabeled data to train the domain discriminator. The optimization goal of the domain discriminator is as follows:

$$\underset{R,ADA}{\operatorname{argmin}} \mathcal{L}(ADA(avg(\vec{x_i})), d_i) \quad (4)$$

We linearly combine the classification loss and the ADA loss to optimize the model. We use the reverse of the ADA loss as we want to penalize the model when it identifies the domain correctly. We do this because the goal of the ADA component is to remove domain-specific features from the representations of M, so that it learns about offensiveness but not in the context of any particular data set.

$$\underset{R,LSTM,C,ADA}{\operatorname{argmin}} \mathcal{L}(C([h_f, h_b]), y_i) - \lambda \mathcal{L}(ADA(avg(\vec{x_i})), d_i) \quad (5)$$

Both loss functions for classification and domain identification are cross-entropy functions that compare the one hot vector from the ground truth with the probability distribution over the classes or data sets computed using a softmax function. Following the work of Ganin and Lempitsky (2015), we implemented ADA using a gradient reversal layer (GRL). During the neural network weight updates (backward pass) the GRL scales the gradients by a factor of λ.

4.2 Difficulty Measurer

The goal of the difficulty measurer is to estimate the similarity between a source training instance from S and a subset of instances from the target domain T on which our model M makes predictions with more certainty. To estimate the similarity, we construct two vector-based pseudo queries for the target domain T, one for the offensive class and one for the neutral class. The similarity of any instance from the source domain S to the two vector-based queries can be calculated by embedding the source instances in the same vector space as the pseudo queries in T.

More specifically, we estimate the prediction uncertainties of the instances in T using a checkpoint of the model at epoch i. The model checkpoint M_i makes predictions over the instances of T, with output probability confidence scores $P(x_j^T = \text{offensive}) = M_i(x_j^T)$, which are used to assign a predicted class for each target instance[1]. Then, for each predicted class, we sort the target instances assigned to that class in the descending order of confidence scores to create a list ranked from higher to lower certainty.

Recall that our model has a transformer backbone, where each target instance can be represented using its corresponding average token embeddings from the output of the model, denoted as R. To create a pseudo vector-based query for the a given (offensive, neutral) class, we take the top-k target instances in the corresponding predicted class, and average the representations R over the k instances. The result is two pseudo vector-based queries in T representing the two classes. For each of these queries, we compute the similarity scores of the instances in S, which constitutes the input for the training scheduler. The scheduler uses these similarity scores to select instances from S to train M in the next epoch, $i + 1$.

Note that in the first epoch, the parameters of the classification model M_0 are randomly initialized (except for the pre-trained layers). Thus, we assume that the predictions and confidence scores from M_0 are not useful and in the first epoch, we assign the same confidence scores to all the target domain unlabeled instances, i.e., we randomly sample from target domain unlabeled instances. We use all the unlabeled training instances to create a single query, find similar examples in the source-domain training data, and use a similarity threshold to select a subset of them to train our randomly initialized model M_0.

To improve the expressiveness of the representations, we augment and extend the dense semantic representations R with the Hurtlex lexicon (Bassignana, Basile, and Patti 2018) for similarity computation. As we mentioned in the Related Work, Hurtlex vectors have proved to be useful for cross-domain offensive language generalization. Following (Koufakou et al. 2020), we construct a one-hot vector that represents the categories of words that appear in an arbitrary sentence. We consider Hurtlex as a key-value store, where offensive terms are stored against a specific type of offensiveness such as racial offensiveness. There are 17 different types of offensiveness in Hurtlex thus we initialize a 17-dimensional one-hot vector of zeros for each of the instances. To update the one-hot vector for an instance, for each token in the instance we check whether there is mapping from that token to Hurtlex tokens. If we find a mapping, we turn the corresponding dimension of the vector to 1.

Following the construction steps, we obtain Hurtlex vectors for all instances from every source- and target-domain data set and store them. We score all the source-domain examples using cosine similarity (Reimers and Gurevych

[1] Following a typical binary classification approach, we classify x_j^T as offensive if $P(x_j^T = \text{offensive}) > 0.5$ and set $P_{\text{offensive}}$ as the confidence score of x_j^T. If x_j^T is predicted to be neutral, we set its confidence score as $1 - P_{\text{offensive}}$.

2019) with both the model M_i and Hurtlex representations of target domain predicted classes, and average the score.

4.3 Training Scheduler

The Training Scheduler component for MultiFOLD takes in difficulty scores of each of the examples in S at a given epoch and decides which of the instances will be used to train M in the next epoch. Once we obtain the scores for all the instances in S as described above, we select a subset of them based on a similarity threshold t. We use the same threshold t for both pseudo-queries. Thus, the number of offensive and neutral instances selected from S depends on t.

To allow M to access more training data from S, following the easy-to-hard instance selection in CL, we use a decay factor, $0 < d < 1$ to decrease the similarity threshold t as we progress through the epochs.

At each epoch, we relax the similarity threshold by multiplying it with the decay factor to allow more instances from S to move into training. Thus, at each epoch, we allow more distant examples from the target domain to enter into the training process. Our assumption is that the more distant a source instance is away from the target, the harder it is for the model to classify it. Our threshold and decay factor captures this assumption, and we show the effectiveness of these in our experimental results section.

5 Experiments

We begin by outlining the experimental setup and datasets. We then present several baselines as a point of comparison for MultiFOLD. Next, we evaluate MultiFOLD and conduct an ablation study to assess the importance of various components of the framework.

5.1 Experimental Setup

Datasets. We evaluate existing baselines for multi-source domain adaptation and our approach using six publicly available datasets. The datasets contain examples from different social media platforms: Twitter, Facebook, Youtube, and StormFront (an online forum dedicated to White supremacy), and contain different types of offensiveness leading to multiple class labels. In our experiments, we address a binary classification task, where we consolidate all types of offensive content into a single positive label without making a distinction across the types. We then label the rest of the instances as non-offensive (i.e. neutral).

A summary of the datasets' characteristics is reported in Table 2. All datasets are partitioned into train (80%), test (10%) and validation (10%) sets in a stratified way taking the class distribution into account. Then we conduct leave-one-dataset-out experiments, where we take out one of the datasets as the target domain dataset and use the remaining datasets as source domains. Here, the training splits of the source domains are combined into a single virtual training set, and the same is done for the validation split.

To set up training, we leverage the labels of source training instances to optimize for the classification loss in Equation 3 wherever applicable. To optimize for the adversarial loss in Equation 4, we use the textual content of source domain and the target domain training set without labels. We emphasize that the labels of the target training instances are not used to guide the model; the only information we leverage here is the textual semantics. In order to evaluate and compare all the approaches, we use the test split from the target domain. We describe all the data sets used in our experiments in the following paragraphs.

SEMEVAL: This dataset was released as part of SemEval 2019 (Basile et al. 2019) challenge for multilingual hate speech detection [2].

SIGIR: This dataset was constructed by merging two previously published Twitter datasets (Davidson et al. 2017; Waseem and Hovy 2016). One of them (Waseem and Hovy 2016) contained racist and sexists tweets while the other (Davidson et al. 2017) contained offensive language in general without an explicit categorization.

WOAH: This dataset (Vidgen et al. 2020) contains Tweets expressing hostility and offensiveness against Asia and Asian people during the COVID-19 pandemic.

ALW2: The instances of this dataset (de Gibert et al. 2018) are sentences collected from Stormfront, a white supremacist forum. Another particular characteristic of this dataset is its class imbalance: only 9% of the data set is labeled offensive.

HASOC: This dataset was part of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages (Mandl et al. 2019). The theme of the datasets could be described as political, with several offensive examples regarding political figures, elections and goverment.

ICWSM: This dataset (Salminen et al. 2018) with only 3222 examples is the smallest dataset in our collection. The comments were recovered from Facebook and Youtube and include different types of offensiveness including racist and xenophobic examples.

Hyperparameters: As we conduct a leave-one-dataset-out experiment, we perform hyperparameter search for each left-out data set , and report the test results based on the optimal hyperparameters for that data set. To obtain evaluation metrics for the test split of a left-out data set we apply grid search over learning rates $\in$ {2e-05, 5e-05}, and batch size $\in$ {16, 32, 64} to search for the optimal hyper-parameters using the validation set of that data set. We train the models for six epochs on a single NVIDIA V100 GPU and select the best model checkpoint based on validation set performance. Note that we derived the hyperparameter spaces from the original BERT paper (Devlin et al. 2018).

We set the LSTM input embedding size, number of time steps, and hidden layer embedding size as 768 (the size of the BERT vectors), 30, and 256, respectively. We use the SGD optimizer with a momentum equal to 0.5. To obtain robust results, we report the average of the evaluation metrics after running all the models with three different random seeds. We did not observe significant empirical differences

[2]The "Multilingual detection of hate speech against immigrants and women in Twitter" task was defined for English and Spanish languages, we use in this work the English portion of the dataset.

Dataset	Type of offensiveness	Platform	Offensive	Not-Offensive	Total
SEMEVAL (Basile et al. 2019)	Sexism, Xenophobia	Twitter	5462	7508	12970
WOAH (Vidgen et al. 2020)	Hostility against Asians	Twitter	5331	14669	20000
SIGIR (Arango, Pérez, and Poblete 2019)	Sexism, racism.	Twitter	2920	4086	7006
ALW2 (de Gibert et al. 2018)	-	StormFront	1196	9748	10944
HASOC (Mandl et al. 2019)	-	Facebook,Twitter	2261	3591	5852
ICWSM (Salminen et al. 2018)	-	Facebook,Youtube	2364	858	3222

Table 2: Summary of characteristics of the data sets used in our evaluation. For each dataset, we show the type of offensiveness if it is described in the original paper, the source platform where the data was extracted from, the number of instances per class, and the total number of instances.

	SEMEVAL	WOAH20	SIGIR19	ALW2	HASOC	ICWSM
SEMEVAL	**64.41**	52.86	56.04	57.12	55.66	49.44
WOAH20	36.68	**72.27**	36.93	68.17	53.54	49.17
SIGIR19	56.05	42.3	**82.6**	53.75	52.38	37.09
ALW2	36.68	59.36	47.13	**80.34**	49.5	21.07
HASOC	42.6	42.35	36.84	45.54	**65**	36.36
ICWSM	29.6	21.05	59.74	38.35	55.38	**77.04**

Table 3: Results on in-domain and cross-domain evaluation. Each column represents a different test set and each row a training set. The diagonal numbers are from in-domain training and evaluation which are the upper bound for MultiFOLD performance.

after a small number of epochs.

Evaluation metrics: We report the experimental results using macro F1-scores on test data. The results are presented as the average performance of the models trained with three different random seeds: 42, 1, and 2022.

5.2 Baselines

Our main hypothesis is that by strategically incorporating training data from multiple sources, the performance of the learned offensiveness detection model can be enhanced in a target domain. To test this hypothesis, we first introduce several baselines to which we compare the detection F1-score using target domain test data. We start with simple baselines such as *single-source classification* to validate that in-domain classifiers perform the best, *multi-source classification* to confirm that randomly adding more data into training process only improves the performance slightly in some cases and does not show a significant improvement over all the target domains, and *self-training* (Triguero, García, and Herrera 2015) to show that adding more data into training process without any knowledge from target domain does not improve the performance compared to multi-source classification. We then evaluate the SOTA work for cross domain following Ryu, Lee, and Lee (2022), adapted to our problem setting (referred to here as BERT-ADA). Our additional baseline is to evaluate the performance of large language models in offensive detection setting to validate the idea that re-training a smaller model with task-specific instances (offensive content in our case) outperforms large (but generic) language models.

In-domain classification. We benchmark in-domain classification results, as well as the results of cross-domain classification without adaptation. For the former, we leverage the data splits from the same domain for training and evaluation. For the latter, a trained model using the training data from a source domain is evaluated on the test data of a target domain directly. In this case, the source and target domains are selected to be different, enabling us to study the effect of out-of-domain classification. The performance results are shown in Table 3. As expected, in-domain classifiers outperform cross-domain classifiers. Note that the diagonal numbers are from in-domain training and evaluation and are considered as the upper bound for MultiFOLD performance.

Multi-source classification. We further study the baseline that uses multiple source domain instances for training rather than a single one. Our hypothesis is that data instances from different domains could be complementary to each other, and overall enhances the detection accuracy in an unseen target domain when combined together.

To implement this, for each target domain T, we combine the training instances from all other source domains S for learning, where the ordering of training instances is randomly shuffled. The corresponding trained model is then used for evaluation on T without adaptation.

Table 4 row labeled *'Multi-source'* shows that including multiple sources improves the out-of-domain performance compared to single source training but still underperforms in-domain classification results. Taking the SEMEVAL dataset as an example for the target domain, the performance of the classifier trained on each of other datasets, namely WOAH20, SIGIR19, ALW2, HASOC and ICWSM, is 36.68, 56.05, 36.68, 42.6 and 29.6 respectively. Compared to those, the performance of the classifier trained using all other sources combined is 59.32. This suggests that leveraging multiple sources could be more effective than using just a single one. However, comparing to the in-domain

classification result of 64.41, we note that there is still room for improvement when using multiple sources, leading us to the hypothesis that strategically combining training instances could be important.

Self-training. In this baseline, we intend to use a self-taught confidence score indicator as a strategy to leverage training instances from multiple sources (Triguero, García, and Herrera 2015). We investigate this approach to understand whether augmenting the training data without knowledge of the target domain improves results. Similar to Multi-source classification, all the training sources S are concatenated except the one from the target domain T. The training process is conducted by taking batches from S in a self-training manner as follows. We first instantiate a classifier using the Multi-source classification approach. We then subsequently take the confidence scores on the same samples in S using the initial model, and incrementally incorporate those with high confidence to update the model. Table 4 row labeled *'Self-training'* shows that including training samples in self-training fashion does not improve the performance in comparison with multi-source classification. Taking the SEMEVAL dataset as an example for the target domain, the performance the classifier in multi-source classification is 59.32. However, when employing the self-training strategy, its performance experiences a slight decrease to 58.7.

BERT-ADA. Ryu, Lee, and Lee (2022) implemented an approach for knowledge distillation for Bert unsupervised domain adaptation. It was originally employed for sentiment analysis, and the paper describes experiments of transferring from one domain to another domain in predicting the sentiment of e-Commerce product reviews. In their setting, it is assumed there is a single source, and a single target domain. The work introduces an adversarial loss and we adapt this baseline to our problem setting by considering the combination of multiple sources as a single source. The results for BERT-ADA is listed in Table 4, row labeled *'BERT-ADA'*. The impact of adversarial loss on performance as observed in WOAH/ SIGIR/ HASOC datasets is significant. The results improve from 58.32/ 60.61/ 53.44 to 64.39/ 65.51/ 58.91 respectively, compared to multi-source classification. This inspired our proposed MultiFOLD approach to leverage a new loss function in addition to task-specific loss in order to generalize the language-specific attributes between source domains and target domain.

Vicuna-13B Inference.[3] Now we turn to evaluate the impact of large language models on offensive content detection. The Vicuna-13b language model has demonstrated a quality level of 90% compared to ChatGPT, making it a competitive publicly available language model[4]. We evaluate Vicuna-13b on each of the test data sets, reported in Table 4 row labeled *'LLM (Vicuna-13B)'* our results confirm re-training a smaller model with task-specific (offensive content in our case) instances outperforms large but generic language models. Zhu et al. (2023) echoes our findings that off-the-shelf LLMs perform poorly on offensive content detection. Taking the SEMEVAL dataset as an example for the target domain, the classifier's performance in multi-source classification is 59.32. However, when employing the LLM, its performance experiences a significant decrease to 49.56. We evaluate Vicuna-13B on the task of offensive language detection in a zero-shot setting where we design the prompt to be "Classify a social media comment as offensive, or, non-offensive. comment : *sample of social media comment in test data*"

Pre-defined CL. In this baseline, we show the impact of simple pre-defined CL in which training samples are ranked based on their similarity to the target domain in a pre-training process and then batched for training. Table 4, row labeled *'Pre-defined CL'* shows that pre-defined CL rarely improves over random combination of data sources. Taking the SEMEVAL dataset as an example for the target domain, the performance the classifier in multi-source classification is 59.32. However, when employing the Pre-defined CL strategy, its performance experiences a slight decrease to 58.58. On the other hand the performance of HASOC slightly improves from 53.44 to 54.22.

5.3 Evaluation Results

In this section, we show the performance of MultiFOLD in comparison to the baselines by conducting an ablation study over its different components. We start with evaluating the automatic CL strategy in the absence of generalization techniques. This shows the impact integrating multiple sources strategically compared to the random combination of the multiple sources (i.e., multi source classification baseline). We then evaluate the offensiveness generalization technique referring to as Hurtlex. For a fair comparison, we also evaluate the impact of Hurtlex on the baseline models, namely Self-training and Pre-defined CL, which employ a sample selection strategy. Finally, we evaluate the whole framework by incorporating ADA loss that generalizes across multiple domains.

Automatic CL. The approach selects source training samples based on their similarity to the target domain in a dynamic fashion. The similarity is computed based on the dense embedding representations of the source instances and the target instances that are predicted with highest confidence scores.

In Table 4, row labeled *'Automatic CL'* shows the results of Automatic CL. Compared to Pre-defined CL, the results suggest that Automatic CL improves the prediction effectiveness in most cases, leading about 2% to 3% absolute increase in F1-scores across the SEMEVAL, WOAH, SIGIR, and ALW2 datasets. The two exceptions are ICWSM and HASOC. We conjecture this is possibly due to the much smaller size of the two datasets, (3k and 5.8k examples respectively). When there is little data to begin with, the dynamic selection of target samples does not produce samples significantly different from those selected in the pre-defined setting.

Automatic CL + Hurtlex. In Automatic CL, we rely on dense embedding representations as the only features to compute similarity scores. The representations, while effective in representing semantics, do not necessarily represent the lexical attributes of offensive content. To address this,

[3]https://github.com/lm-sys/FastChat

[4]https://lmsys.org/blog/2023-03-30-vicuna/

		SEMEVAL	WOAH	SIGIR	ALW2	HASOC	ICWSM
Baselines	In-domain training	64.41	72.27	82.60	80.34	65.00	77.04
	Multi-source	59.32	58.32	60.61	63.77	53.44	56.47
	Self-training	58.7	52.9	61.1	63.7	50.02	52.2
	Self-training + Hurtlex	60.1	58.3	60.7	62.7	53.8	58.6
	BERT-ADA (Ryu, Lee, and Lee 2022)	58.73	64.39	65.51	59.76	58.91	52.59
	LLM (Vicuna-13B)	49.56	42.54	54.69	47.72	41.53	**68.55**
	Pre-defined CL (Rao 2020)	58.58	58.03	59.48	61.14	54.22	65.45
	Pre-defined CL + Hurtlex	61.91	54.91	63.74	61.31	54.35	61.03
MultiFOLD	Automatic CL	61.87	61.07	62.38	63.83	54.02	59.89
	Automatic CL + Hurtlex	61.66	60.79	66.32	**64.81**	**60.90**	60.00
	Automatic CL + ADA	**61.93**	56.42	65.88	64.77	49.20	49.52
	Automatic CL + Hurtlex + ADA	60.49	**64.73**	**69.82**	62.73	53.79	**68.21**

Table 4: Results on evaluation of the baselines and MultiFOLD. The model is trained using all the datasets with the exception of the target domain. Each row of the table represents a different training strategy and each column represent a different testing set.We report macro F1-scores on test data. The results are presented as the average performance of the models trained with three different random seeds: 42, 1, and 2022.

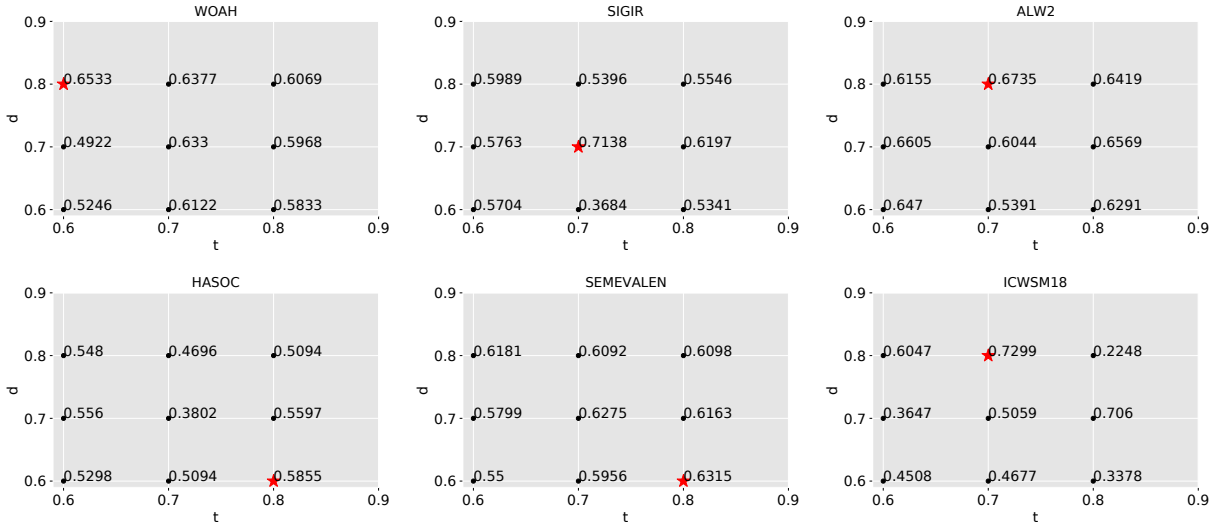

Figure 2: The performance with regard to different values of threshold (t) and decay factor (d) for each of the target data sets. The star symbol shows the best performing combination of t and d.

we incorporate Hurtlex to generalize across domains with distinct offensive vocabularies. More specifically, we concatenate the semantic embeddings and the Hurtlex representations as explained in Section 4.2 and compute similarity over the unified vectors.

The result is shown in Table 4, row labeled *'Automatic CL + Hurtlex'*. Compared to Automatic CL, we note that incorporating Hurtlex improves the classification performance in most cases, suggesting its capability to associate source and target domains through the shared offensiveness lexical features. In particular, an average of 2% absolute increase in F1-score is observed across all datasets, with the most improvements found in SIGIR and HASOC datasets.

We delve deeper into the Hurtlex vocabularies and their intersection with the validation set from the target domain. This analysis aims to gauge the extent to which Hurtlex encompasses the characteristics of the target domain's offensiveness, allowing CL to incorporate this specific type of offensiveness into the training process. In the case of SIGIR, ALW2, HASOC, and ICWSM datasets, approximately 4%, 4.44%, 2.54%, and 5.01% of their validation set vocabulary is found in the Hurtlex lexicons. This finding confirms the beneficial impact of Hurtlex on model performance in these datasets. Conversely, for the SEMEVAL and WOAH datasets, Hurtlex's representation is relatively sparse, leading to no observable improvement in performance. This discrepancy can be attributed to Hurtlex which contains around 8,000 offensive words, primarily generic forms of offensiveness, while WOAH specifically addresses tweets expressing hostility directed towards Asia and Asians during COVID19, which may not be adequately captured by Hurtlex's lexicon.

Baselines + Hurtlex. For a fair comparison, we have also integrated Hurtlex into the baseline models that employ a sample selection strategy. In Table 4, rows labeled *'Self-*

training + Hurtlex' and *'Pre-defined CL + Hurtlex'*, we note improved performance compared to the baseline models. However, the model *'Automatic CL + Hurtlex'* surpasses the performance of the other two models, underscoring the significant impact of Automatic CL in which training instances are strategically sampled.

Automatic CL + ADA. We evaluate the impact of the ADA component employed to generalize the model across language-specific attributes. Briefly, for language adaptation, we add two extra layers to the model (a discriminator) that distinguish the source and target domains. The gradient reverse layer (GRL) back propagates the gradient in an adversarial way so that sources and target become indistinguishable. As shown in Table 4, row labeled *'Automatic CL + ADA'*, the ADA component improves the performance across SEMEVAL, SIGIR, ALW2 compared to the model that solely employs Automatic CL. The exceptions are ICWSM and HASOC. We propose that the model struggles to effectively capture the language attributes specific to these data sets due to their limited size.

Automatic CL + Hurtlex + ADA. Finally, we evaluate MultiFOLD as a whole along with all the proposed components. As shown in Table 4, row labeled *'Automatic CL + Hurtlex + ADA'* , MultiFold performs the best for three target datasets and improves the performance across all target datasets (except HASOC) compared to both SOTA (BERT-ADA) and Pre-defined CL. Specifically, SIGIR and ICWSM datasets improve by 3.5% and 8.21% with the inclusion of the ADA component.

5.4 MultiFOLD Hyper-Parameter Analysis

We propose Automatic CL to select training samples in a strategic manner for offensiveness detection. There are two main hyper-parameters, namely threshold (t) and decay factor (d), that determine the selection of the most effective training samples. MultiFOLD selects training samples based on their similarity to the target domain with respect to t in the first epoch. The decay factor relaxes the similarity threshold over subsequent epochs, selecting more training samples. We deploy a grid search over values of $t \in [0.6, 0.7, 0.8]$ and $d \in [0.6, 0.7, 0.8]$. Figure 2 shows the best performing combination across different target domain data sets.

Note that the target domains with the lowest performance, namely HASOC and SEMEVAL (Table 4), require a larger optimal threshold value ($t = 0.8$) and a smaller decay factor ($d = 0.6$). This suggests that including more training examples in the initial epoch can lead to improved performance, possibly due to the relatively few examples in each data set. The average optimal threshold, ($t = 0.7$), (SIGIR, ALW2 and ICWSM18) assists in selecting an appropriate number of training samples in the initial epoch. It should be noted that hyper-parameter tuning experiments are run over MultiFOLD without the Hurtlex or ADA components.

6 Broader Perspective

The main data used in this work – offensive and non-offensive instances from different social media platforms – are publicly accessible data. We did not collect or use any personal identifiable information that can identify the author of social media posts. This work highlights the value of textual information shared on social media for advancing the state of the art in automatically detecting offensive content, while decreasing annotation effort. This reduces the chances of harm to people (including annotators) who do not wish to see offensive content.

Though our proposed approach leverages multiple source datasets to optimize performance on a target dataset, we reduced the label space of these datasets to a binary label to simplify the problem. It will require non-trivial modifications to our framework to adapt it to a target dataset where the label space is complex and fine-grained. In addition, our dataset collection may risk centering on certain types of offensive content or specific ways that offensiveness is expressed, since a big proportion of our data originate from a few social platforms. We anticipate that such problems could be alleviated by progressively adding more diverse corpora, such as multi-lingual or long-form textual content.

The goal of this work is to automatically detect and avoid offensive content. While unlikely, it is not impossible that bad actors could employ the approach to discover unlabelled offensiveness from a new corpus to cause additional harm (e.g. purposely disseminating or replicating the material). Our recommendation is for content filtering providers to proactively assess new content and mark high-risk instances for further analysis.

7 Conclusion

This paper presented the MultiFOLD framework which adapts offensive language detection from multiple source domains to an unseen target domain. Extensive experiments on six different target domains were compared to recent strong baselines. The experimental results suggest that following an easy-to-hard dynamic curriculum learning (Automatic CL) effectively adapts observations from multiple source domains, compared to a strategy of pre-defining the easy-to-hard ranking of examples. We found that augmenting text representations by incorporating Hurtlex and domain adversarial training improves the estimation of similarity, leading to more effective instance selection. By orchestrating between the Difficulty Measurer and the Training Scheduler, our approach was able to gradually train the offensive classifier to adapt to a target domain.

While we have selected training instances from multiple source domains, in future work we anticipate that leveraging domain-level properties aggregated could enhance the instance selection. A related direction we hope to investigate further, is to model the difficulty of each source data set with respect to the specific variants of offensive language.

References

Aluru, S. S.; Mathew, B.; Saha, P.; and Mukherjee, A. 2020. Deep Learning Models for Multilingual Hate Speech Detection. *CoRR*, abs/2004.06465.

Arango, A.; Pérez, J.; and Poblete, B. 2019. Hate Speech Detection is Not as Easy as You May Think: A Closer Look

at Model Validation. In Piwowarski, B.; Chevalier, M.; Gaussier, É.; Maarek, Y.; Nie, J.; and Scholer, F., eds., *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, 45–54. ACM.

Arora, A.; Nakov, P.; Hardalov, M.; Sarwar, S. M.; Nayak, V.; Dinkov, Y.; Zlatkova, D.; Dent, K.; Bhatawdekar, A.; Bouchard, G.; and Augenstein, I. 2023. Detecting Harmful Content on Online Platforms: What Platforms Need vs. Where Research Efforts Go. *ACM Comput. Surv.*, 56(3).

Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Pardo, F. M. R.; Rosso, P.; and Sanguinetti, M. 2019. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In May, J.; Shutova, E.; Herbelot, A.; Zhu, X.; Apidianaki, M.; and Mohammad, S. M., eds., *Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019*, 54–63. Association for Computational Linguistics.

Bassignana, E.; Basile, V.; and Patti, V. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, 1–6. CEUR-WS.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Bose, T.; Aletras, N.; Illina, I.; and Fohr, D. 2022a. Domain Classification-based Source-specific Term Penalization for Domain Adaptation in Hate-speech Detection. *arXiv preprint arXiv:2209.08681*.

Bose, T.; Aletras, N.; Illina, I.; and Fohr, D. 2022b. Dynamically Refined Regularization for Improving Cross-corpora Hate Speech Detection. *arXiv preprint arXiv:2203.12536*.

Bose, T.; Illina, I.; and Fohr, D. 2021a. Generalisability of Topic Models in Cross-corpora Abusive Language Detection. In *NLP4IF 2021-Workshop Censorship, Disinformation, and Propaganda*.

Bose, T.; Illina, I.; and Fohr, D. 2021b. Unsupervised domain adaptation in cross-corpora abusive language detection. In *SocialNLP 2021-The 9th International Workshop on Natural Language Processing for Social Media*.

Caselli, T.; Basile, V.; Mitrović, J.; and Granitzer, M. 2020. Hatebert: Retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

Chatzakou, D.; Kourtellis, N.; Blackburn, J.; Cristofaro, E. D.; Stringhini, G.; and Vakali, A. 2017. Mean Birds: Detecting Aggression and Bullying on Twitter. In *Proceedings of the 2017 ACM on Web Science Conference, WebSci 2017, Troy, NY, USA, June 25 - 28, 2017*, 13–22.

Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; and Villata, S. 2019. Cross-platform evaluation for Italian hate speech detection. In *CLiC-it 2019-6th Annual Conference of the Italian Association for Computational Linguistics*.

Davidson, T.; Bhattacharya, D.; and Weber, I. 2019. Racial Bias in Hate Speech and Abusive Language Detection Datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, 25–35.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, 512–515.

de Gibert, O.; Pérez, N.; Pablos, A. G.; and Cuadros, M. 2018. Hate Speech Dataset from a White Supremacy Forum. In Fiser, D.; Huang, R.; Prabhakaran, V.; Voigt, R.; Waseem, Z.; and Wernimont, J., eds., *Proceedings of the 2nd Workshop on Abusive Language Online, ALW@EMNLP 2018, Brussels, Belgium, October 31, 2018*, 11–20. Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gambäck, B.; and Sikdar, U. K. 2017. Using Convolutional Neural Networks to Classify Hate-Speech. In *Proceedings of the First Workshop on Abusive Language Online*, 85–90. Association for Computational Linguistics.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, 1180–1189. PMLR.

Golbeck, J. 2023. Misogyny, Women in Power, and Patterns of Social Media Harassment. In Thomson, R.; Alkhateeb, S.; Burger, A.; Park, P.; and A. Pyke, A., eds., *Social, Cultural, and Behavioral Modeling*, 3–11. Springer Nature Switzerland. ISBN 978-3-031-43129-6.

Guo, H.; Pasunuru, R.; and Bansal, M. 2020. Multi-source domain adaptation for text classification via distancenet-bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7830–7838.

Karan, M.; and Šnajder, J. 2018. Cross-domain detection of abusive language online. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*, 132–137.

Koufakou, A.; Pamungkas, E. W.; Basile, V.; Patti, V.; et al. 2020. HurtBERT: incorporating lexical features with BERT for the detection of abusive language. In *Fourth Workshop on Online Abuse and Harms*, 34–43. Association for Computational Linguistics.

Liu, J.; Zheng, S.; Xu, G.; and Lin, M. 2021. Cross-domain sentiment aware word embeddings for review sentiment analysis. *International Journal of Machine Learning and Cybernetics*, 12(2): 343–354.

Ludwig, F.; Dolos, K.; Zesch, T.; and Hobley, E. 2022. Improving Generalization of Hate Speech Detection Systems to Novel Target Groups via Domain Adaptation. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, 29–39.

Mandl, T.; Modha, S.; Majumder, P.; Patel, D.; Dave, M.; Mandalia, C.; and Patel, A. 2019. Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages. In Majumder, P.; Mitra, M.; Gangopadhyay, S.; and Mehta, P., eds., *FIRE '19: Forum for Information Retrieval Evaluation, Kolkata, India, December, 2019*, 14–17. ACM.

Mosallanezhad, A.; Karami, M.; Shu, K.; Mancenido, M. V.; and Liu, H. 2022. Domain Adaptive Fake News Detection

via Reinforcement Learning. In *Proceedings of the ACM Web Conference 2022*, 3632–3640.

Pamungkas, E. W.; Basile, V.; and Patti, V. 2020. Misogyny detection in twitter: a multilingual and cross-domain study. *Information Processing & Management*, 57(6): 102360.

Pamungkas, E. W.; Basile, V.; and Patti, V. 2021. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing & Management*, 58(4): 102544.

Pamungkas, E. W.; and Patti, V. 2019. Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual meeting of the association for computational linguistics: Student research workshop*, 363–370.

Plaza-Del-Arco, F. M.; Molina-González, M. D.; Ureña-López, L. A.; and Martín-Valdivia, M. T. 2021. A multi-task learning approach to hate speech detection leveraging sentiment analysis. *IEEE Access*, 9: 112478–112489.

Rao, V. A. e. a. 2020. A sentiwordnet strategy for curriculum learning in sentiment analysis. In *International Conference on Applications of Natural Language to Information Systems*, 170–178. Springer.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ryu, M.; Lee, G.; and Lee, K. 2022. Knowledge distillation for bert unsupervised domain adaptation. *Knowledge and Information Systems*, 64(11): 3113–3128.

Salminen, J.; Almerekhi, H.; Milenkovic, M.; Jung, S.; An, J.; Kwak, H.; and Jansen, B. J. 2018. Anatomy of Online Hate: Developing a Taxonomy and Machine Learning Models for Identifying and Classifying Hate in Online News Media. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*, 330–339. AAAI Press.

Sarwar, S. M.; and Murdock, V. 2022. Unsupervised domain adaptation for hate speech detection using a data augmentation approach. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, 852–862.

Sarwar, S. M.; Zlatkova, D.; Hardalov, M.; Dinkov, Y.; Augenstein, I.; and Nakov, P. 2022. A neighborhood framework for resource-lean content flagging. *Transactions of the Association for Computational Linguistics*, 10: 484–502.

Soviany, P.; Ionescu, R. T.; Rota, P.; and Sebe, N. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 1–40.

Sun, B.; Feng, J.; and Saenko, K. 2016. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Triguero, I.; García, S.; and Herrera, F. 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42: 245–284.

Vidgen, B.; Hale, S. A.; Guest, E.; Margetts, H. Z.; Broniatowski, D. A.; Waseem, Z.; Botelho, A.; Hall, M.; and Tromble, R. 2020. Detecting East Asian Prejudice on Social Media. In Akiwowo, S.; Vidgen, B.; Prabhakaran, V.; and Waseem, Z., eds., *Proceedings of the Fourth Workshop on Online Abuse and Harms, WOAH 2020, Online, November 20, 2020*, 162–172. Association for Computational Linguistics.

Wang, X.; Chen, Y.; and Zhu, W. 2022. A Survey on Curriculum Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 4555–4576.

Waseem, Z.; and Hovy, D. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 88–93.

Wiegand, M.; Ruppenhofer, J.; and Kleinbauer, T. 2019. Detection of abusive language: the problem of biased datasets. In *Proceedings of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: human language technologies, volume 1 (long and short papers)*, 602–608.

Yang, L.; Balaji, Y.; Lim, S.-N.; and Shrivastava, A. 2020. Curriculum manager for source selection in multi-source domain adaptation. In *European Conference on Computer Vision*, 608–624. Springer.

Zhan, R.; Liu, X.; Wong, D. F.; and Chao, L. S. 2021. Meta-curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14310–14318.

Zhang, X.; Shapiro, P.; Kumar, G.; McNamee, P.; Carpuat, M.; and Duh, K. 2019. Curriculum learning for domain adaptation in neural machine translation. *arXiv preprint arXiv:1905.05816*.

Zhao, S.; Xiao, Y.; Guo, J.; Yue, X.; Yang, J.; Krishna, R.; Xu, P.; and Keutzer, K. 2021. Curriculum cyclegan for textual sentiment domain adaptation with multiple sources. In *Proceedings of the Web Conference 2021*, 541–552.

Zhu, Y.; Zhang, P.; Haq, E.-U.; Hui, P.; and Tyson, G. 2023. Can ChatGPT Reproduce Human-Generated Labels? A Study of Social Computing Tasks. *arXiv preprint arXiv:2304.10145*.

Paper Checklist

1. For most authors...

   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? The paper improves the state of the art in offensive language detection, which seeks to minimize harm to societies, cultures, and individuals. The data used does not contain personal information, and is in the public domain.

   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope?

The abstract summarizes the proposed approach MultiFOLD and the experimental results of the work. The introduction expands the abstract by providing the motivation and highlighting the innovations of MultiFOLD compared to prior art. Both the abstract and introduction accurately reflect the work's contributions and scope.

(c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes. We described in depth the design of our approach to addressing the problem of interest. Following that, we supported our claims and demonstrated the efficacy of the approach through extensive experiments, including comparison with eight baselines on six public datasets and studying the effects of the components in MultiFOLD.

(d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? The datasets are cited, and described in this work. We did not do additional analysis of demographic distributions represented in the data. Instead we cited and discussed in the related work section prior work using the same data, that identifies artifacts in the data (i.e. Arango, Pérez, and Poblete (2019)).

(e) Did you describe the limitations of your work? In section 6, we discussed the limitations of our work, along with potential directions for mitigation.

(f) Did you discuss any potential negative societal impacts of your work? The work intends to uncover offensive content without extensive human labeling. In section 6, we reiterated how the proposed approach can reduce negative societal impacts for both consumers and annotators.

(g) Did you discuss any potential misuse of your work? In section 6, we discussed how bad actors may leverage the proposed approach to uncover offensiveness for malicious purposes. It is our recommendation that platform providers to proactively analyze new content and enhance the guardrails progressively.

(h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? This work does not release or leverage PII data, relieving us from privacy concerns if any. The steps to reproduce the framework were described in detail through out Sections 4 and 5, while the datasets used are publicly accessible.

(i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.

2. Additionally, if your study involves hypotheses testing...

(a) Did you clearly state the assumptions underlying all theoretical results? NA

(b) Have you provided justifications for all theoretical results? NA

(c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA

(d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA

(e) Did you address potential biases or limitations in your theoretical framework? NA

(f) Have you related your theoretical results to the existing literature in social science? NA

(g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA

3. Additionally, if you are including theoretical proofs...

(a) Did you state the full set of assumptions of all theoretical results? NA

(b) Did you include complete proofs of all theoretical results? NA

4. Additionally, if you ran machine learning experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Releasing code requires a lengthy approval process which would significantly delay the publication of the paper. The algorithm implementation is straightforward, so we opted to publish the paper without releasing the code.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes, in Section 5.1, with a hyperparameter analysis in Section 5.4

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? No. However, we reported the average F1-scores across three random seeds in order to provide a more robust and reliable interpretation of the experimental results.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes. We stated in Section 5.1 that all the experiments were conducted on a single NVIDIA V100 GPU.

(e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes.

(f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? No.

5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? Yes.

(b) Did you mention the license of the assets? Licenses, where applicable, are mentioned in the cited sources.

(c) Did you include any new assets in the supplemental material or as a URL? NA

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? In the Broader Perspectives section we mention that no PII is present in the data. The paper is about offensive language detection, so the data by design contains offensive content.

(f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see **?**)? NA

(g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see **?**)? NA

6. Additionally, if you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots? NA

(b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA

(d) Did you discuss how data is stored, shared, and deidentified? NA