Domain Classification-based Source-specific Term Penalization for Domain Adaptation in Hate-speech Detection

Tulika Bose, Nikolaos Aletras, Irina Illina, Dominique Fohr

To cite this version:

Tulika Bose, Nikolaos Aletras, Irina Illina, Dominique Fohr. Domain Classification-based Source-specific Term Penalization for Domain Adaptation in Hate-speech Detection. COLING 2022 - Proceedings of the 29th International Conference on Computational Linguistics, Oct 2022, Gyeongju, South Korea. hal-03815708

HAL Id: hal-03815708
https://inria.hal.science/hal-03815708
Submitted on 14 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Domain Classification-based Source-specific Term Penalization for Domain Adaptation in Hate-speech Detection

Tulika Bose † Nikolaos Aletras ‡ Irina Illina † Dominique Fohr †
† Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
‡ University of Sheffield, United Kingdom
{tulika.bose, illina, dominique.fohr}@loria.fr
n.aletras@sheffield.ac.uk

Abstract

Warning: this paper contains content that may be offensive and distressing.

State-of-the-art approaches for hate-speech detection usually exhibit poor performance in out-of-domain settings. This occurs, typically, due to classifiers overemphasizing source-specific information that negatively impacts its domain invariance. Prior work has attempted to penalize terms related to hate-speech from manually curated lists using feature attribution methods, which quantify the importance assigned to input terms by the classifier when making a prediction. We, instead, propose a domain adaptation approach that automatically extracts and penalizes source-specific terms using a domain classifier, which learns to differentiate between domains, and feature-attribution scores for hate-speech classes, yielding consistent improvements in cross-domain evaluation.

1 Introduction

While recent state-of-the-art hate-speech classifiers (Ayo et al., 2021; D’Sa et al., 2020; Mozafari et al., 2019) yield impressive performance on in-domain held-out instances, they suffer when evaluated on out-of-domain settings (Yin and Zubiaga, 2021; Arango et al., 2019; Swamy et al., 2019; Karan and Šnajder, 2018). The distributions across corpora/domains change due to varying vocabulary, topics of discussion over time (Florio et al., 2020; Saha and Sindhwani, 2012), data bias caused by sampling strategies (Wiegand et al., 2019) and different hate-targets. This is concerning since curating new data resources for hate-speech involves substantial time and effort (Poletto et al., 2019; Malmasi and Zampieri, 2018). This calls for strategies, like Domain Adaptation (DA) approaches, that can adapt models trained on existing labeled resources to a new target domain that lacks class-labels.

However, research on DA in hate-speech is limited (Sarwar and Murdock, 2022; Bashar et al., 2021; Bose et al., 2021). Typically, vanilla classifiers tend to learn more from domain-specific features (Ye et al., 2021; Wiegand et al., 2019) than domain-invariant features, resulting in poor out-of-domain performance. For instance, Wiegand et al. (2019) show that in a hate-speech dataset (Waseem and Hovy, 2016), neutral domain-specific terms, like ‘football’, ‘commentator’, etc., discussing the role of women in sports, are highly correlated with the hate label, restricting its generalizability. Thus, it is worth minimizing the importance of such terms for improving cross-domain performance.

Recently, feature attributions—methods for extracting post-hoc model explanations, have been used to align features with prior domain knowledge (Rieger et al., 2020; Adebayo et al., 2020). These provide importance scores to the input terms as per their contribution towards the model prediction (Lundberg and Lee, 2017). For instance, Liu and Avci (2019); Kennedy et al. (2020) reduce the oversensitivity of classifiers on a curated list of identity terms (e.g. muslims, gay) by penalizing their importance. However, newly emerging social-media terms (Grieve et al., 2018) may render such lists non-exhaustive. Yao et al. (2021) do not use any list but they require human-provided refinement advice as inputs. Chrysostomou and Aletras (2022a) further show that post-hoc explanation methods might not provide faithful explanation in out-of-domain settings. The contemporaneous work by Attanasio et al. (2022) and Bose et al. (2022) reduce lexical overfitting automatically with entropy-based attentions and feature attributions, respectively. While cross-domain classification performance across different datasets is not studied in the former, the latter needs some labeled target instances to identify the over-fitted terms.

In the task of detecting objects in images, Zunino et al. (2021) use a domain classifier, trained to dif-

1 We use the terms ‘corpus’ and ‘domain’ interchangeably.
ferentiate between domains, to visually identify the irrelevant background information to be domain-specific. Thus, they enforce the model explanations to align with the ground-truth annotations highlighting the objects in the image. Inspired by this, we propose a new DA approach in hate-speech employing a domain classifier, but without having access to such annotations for aligning the attribution scores.

We hypothesize that domain-specific terms that are simultaneously predictive of the hate-speech labels are instrumental in restricting the domain invariance of the hate-speech classifier. To this end, we employ a domain classifier to automatically extract the terms that help in identifying the source domain compared to the unlabeled target domain, and feature-attribute scores to identify the subset important for hate-speech classification from the source. Our method, through penalization of these terms, automatically enforces the source domain classifier to focus on domain-invariant content. Compared to approaches transforming high-dimensional intermediate representations to reduce the domain discrepancy, such as domain adversarial learning (Ryu and Lee, 2020; Tzeng et al., 2017), our approach makes the adaptation more explainable, while improving the overall cross-domain performance compared to prior-approaches.

2 Proposed Approach

Given training data from a labeled source domain $D_S^{train}$ and an unlabeled target domain $D_T^{train}$, our approach for DA in hate-speech involves 2 steps: (i) extraction of source-specific terms and (ii) reducing the importance of these terms. Our setting is similar to Ben-David et al. (2020) and Ryu and Lee (2020).

2.1 Extraction of Source-specific Terms

Domain classification To identify source-specific terms, we first train a binary domain classifier using $D_S^{train}$ and $D_T^{train}$ that learns to identify whether a candidate instance comes from the source or the target domain. For this, we use a simple Logistic Regression (LR) with bag-of-words, as it is inherently interpretable. We then use its feature weights to extract the top $N$ most important terms for predicting the source domain class. Each term is tokenized with the BERT (Devlin et al., 2019) WordPiece tokenizer for compatibility with transformer models. The top $N$ terms obtained through domain classification are denoted as $S_{LR}$.

Attribution-based term ranking Intuitively, the terms from $S_{LR}$ that also contribute highly to the hate-speech labels, are likely to restrict generalization to the target as they could potentially reduce the importance assigned by the classifier to domain-invariant hate-speech terms. Thus, we extract only those source-specific terms that are highly correlated with the labels, given the binary classification task of hate versus non-hate.

To this end, we first continue pre-training BERT on the unlabeled $D_T^{train}$ using the Masked Language Model (MLM) objective for incorporating the language-variations of the target domain, following Glavaš et al. (2020). We then perform supervised classification on $D_T^{train}$ using this MLM trained model. After every epoch, we obtain 2 ranked lists of terms for the two classes, sorted in the order of decreasing importance. We construct the lists using feature attribution methods that yield instance-level attribution scores $\text{ins-atr}_{te}^j$ per term $te$ in an instance $j$ – a higher score indicating a higher contribution to the predicted class. We discard the scores of stop-words and the infrequent terms, and normalize $\text{ins-atr}_{te}^j$ using the sigmoid function. For obtaining a corpus-level class-specific attribution score $\text{cp-atr}_{te}^c$ per term $te$ and per class $c$, we perform a corpus-level average of all the $\text{ins-atr}_{te}^j$ for every $c$ using Equation 1.

$$
\text{cp-atr}_{te}^c = \frac{\sum_{j=1}^{D_T^{train}} 1_{\hat{y}_j = c} \text{ins-atr}_{te}^j \text{ occurrence of } te \text{ in } j}{\sum_{j=1}^{D_T^{train}} 1_{\hat{y}_j = c} \# \text{ (occurrence of } te \text{ in } j)}
$$

Here $c \in \{\text{hate, non-hate}\}$, $\hat{y}$ is the predicted class and $\mathbb{1}$ is the indicator function. We sort the scores $\text{cp-atr}_{te}^c$ for all $te$ to obtain the highest attributed (i.e. most important) term per class to the lowest, yielding the ranked lists of terms per class, given by $\text{CP} = [\text{cp-hate, cp-non-hate}]$.

We extract the source-specific terms $te^S$ that are common to both $S_{LR}$ and the top $M$ terms from CP, i.e. $te^S = \{te \in S_{LR} \& te \in \text{top}_M(\text{CP})\}$. These steps are repeated after every epoch. Note that the list $S_{LR}$ remains constant across the epochs, as it is independent to the hate-speech classification task.

2.2 Penalization of Source-specific Terms

We hypothesize that penalizing $te^S$ obtained from the previous epoch during the next epoch should reduce the importance of terms that are both (i) domain-specific and (ii) contribute highly to the source labels, and thus, help learn from domain invariant terms. We minimize the attribution scores
for te^S, with L_2 penalization, in Equation 2.

\[ \mathcal{L} = \mathcal{L}' + \lambda \mathcal{L}_{atr} = \sum_{t \in \text{te}^S} \phi(t)^2 ; t \in \text{te}^S \]  

(2)

Here \( \mathcal{L}' \) is the classification loss and \( \mathcal{L}_{atr} \) is the attribution loss. \( \lambda \) controls the strength of penalization, and \( \phi(t) \) is the attribution score for \( t \).

We experiment with two variations: (i) **Dom-spec** penalizing only the terms in \( \text{te}^S \); (ii) **Comb** penalizing the combination of \( \text{te}^S \) and the terms from Liu and Avci (2019); Kennedy et al. (2020).

We use two different feature attribution methods that have been widely used in recent studies (Chrysostomou and Aletras, 2021, 2022b): (i) **Scaled Attention** (\( \alpha \nabla \alpha \)) (Serrano and Smith, 2019), which scales attention weights \( \alpha \) by their corresponding gradients \( \nabla \alpha_i = \frac{\partial y}{\partial \hat{y}_i} \), where \( \hat{y} \) is the predicted label, and is shown to work better than using only the attention weights; (ii) **DeepLIFT/DL** (Shrikumar et al., 2017) that assigns scores based on the difference between activation of each neuron and a reference activation (zero embedding vector). Note that although Liu and Avci (2019) have used the Integrated Gradients (IG) (Sundararajan et al., 2017), we use DL as it is most often a good and a faster approximation of IG (Ancona et al., 2018).

### 3 Experimental Setup

#### 3.1 Data

We use three standard hate-speech datasets, namely, **Waseem** (Waseem and Hovy, 2016), **HatEval** (Basile et al., 2019) and **Vidgen** (Vidgen et al., 2021). Following Wiegand et al. (2019); Swamy et al. (2019), we perform hate/non-hate classification across domains. We use the standard splits available for **HatEval** (42.1% hate; train: 8993\(^2\), val: 1000; test: 3000) and **Vidgen** (54.4% hate; train: 32497, val: 1016, test: 4062). We subsample the **Vidgen** validation set by 25% to get 1016 samples, making its size similar to the other datasets. We split **Waseem** (26.8% hate) randomly into train (80%; 8720), validation (10%; 1090) and test (10%; 1090) sets, as no standard splits are available.

We present the top ten most frequent terms in these datasets in Table 1. The **Waseem** dataset is known to comprise a high proportion of implicit hate (Wiegand et al., 2019), which are subtle expressions of hate without the use of profanity. This is also evident in the most frequent terms from this dataset. In Table 1, \#mkr refers to a cooking show which frequently results in sexist comments targeted towards the participating women. **HatEval** involves hate against women and immigrants. Many hateful tweets against immigrants occurred in the context of the US-Mexico border issues with the hashtag #buildthewall. The **Vidgen** dataset is collected through a dynamic data creation process with a human-and-model-in-the-loop strategy, unlike **HatEval** and **Waseem** datasets that are sampled from Twitter. In particular, the **Vidgen** dataset involves hate against many different target groups or identity terms, with a wide variety of topics and hateful forms. See Appendix A for further details on the datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frequent terms in the datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waseem</td>
<td>#mkr, #notsexist, kat, women, like, andre, get, people, one, think</td>
</tr>
<tr>
<td>HatEval</td>
<td>b<em>tch, women, refugees, #buildthewall, immigrant, immigration, illegal, men, migrants, h</em>e</td>
</tr>
<tr>
<td>Vidgen</td>
<td>people, black, women, f*cking, like, love, think, white, get, want</td>
</tr>
</tbody>
</table>

Table 1: Top ten most frequent terms in the datasets after removing the stop-words.

#### 3.2 Baselines

We compare our work with approaches that penalize (i) pre-defined terms in Convolutional Neural Networks-based Liu and Avci (2019)\(^3\); (ii) (a) the identity terms in the top features of a bag-of-words Logistic Regression in BERT-based Kennedy et al. (2020)\(^4\) (b) all the terms listed by Kennedy et al. (2020); (iii) terms extracted automatically by Attanasio et al. (2022); (iv) combination of terms from (i) and (ii,b) within BERT, and call this **Pre-Def**. We do not compare with Bose et al. (2022) as they use labeled target instances for term-extraction, which does not allow a fair comparison.

Further, we experiment with the Vanilla baseline (Van-MLM-FT), where the pre-trained BERT is adapted to \( D^T_{train} \) using the MLM objective, followed by a supervised fine-tuning on \( D^S_{train} \). We also assess different DA methods from the sentiment classification task, namely, **BERT PERL** (Pivot-based Encoder Representation of Language)

---

\(^2\)The instances containing only URLs are removed, decreasing the number of train instances from 9000 to 8993.

\(^3\)with Integrated Gradients (Sundararajan et al., 2017)

\(^4\)with Sampling and Occlusion (Jin et al., 2020)
(Ben-David et al., 2020) that adopts the MLM objective of BERT to perform pivot-based fine-tuning; BERT-AAD (Adversarial Adaptation with Distillation) (Ryu and Lee, 2020) that performs domain adversarial training; HATN (Hierarchical Attention Transfer Network) (Li et al., 2018, 2017) that extracts pivots using a domain adversarial approach.

We evaluate a data-augmentation-based approach (Sarwar and Murdock, 2022) for DA in hate-speech. For a fair comparison, we use the BERT as the underlying model in this approach. Finally, we apply the \( \chi^2 \)-test with 1 degree of freedom and Yates’ correction (Kilgarriff, 2001), penalizing the terms from \( D^{\text{train}}_S \), using their DL scores, for which the null hypothesis of both \( D^{\alpha \text{ain}}_S \) and \( D^{\text{train}}_S \) being random samples of the same larger population, is rejected with 95% confidence. We initialize all the BERT models with MLM adaptation on the target, except for PERL and AAD, which inherently adapts to the target.

### 3.3 Model training

We train all the models on \( D^{\text{train}}_S \), use a small amount of the labeled \( D^{\text{val}}_T \) only for model-selection and hyper-parameter tuning (see Appendix B), following Dai et al. (2020); Maharana and Bansal (2020), and evaluate on \( D^{\text{test}}_T \).

### 4 Results

#### 4.1 Discussion

Table 2 displays the macro-F1 scores obtained, in cross-domain settings, averaged across five randomly initialized runs. We use macro-F1 as penalizing \( e^S \) corrects the mis-classifications for both the hate and non-hate classes across domains. We observe an overall performance drop, compared to Van-MLM-FT, with the DA approaches, originally proposed for sentiment classification, namely, BERT PERL, BERT-AAD and HATN. This also agrees with Bose et al. (2021), who analyze the extracted pivots – terms that are both frequent across domains as well as important for classification with respect to the source – and find them to be sub-optimal for DA in hate-speech. The approach by Sarwar and Murdock (2022) also displays an overall drop. They augment the source domain by substituting relevant terms from a different negative emotion dataset with tagged hate-speech related terms from the target domain. We observe that the augmented instances are often in comprehensible after such substitution.

Dom-spec yields improvements over all the baselines using both \( \alpha \nabla \alpha \) and DL, both independently and in combination (Comb) with Pre-def, where Comb achieves the highest overall performance with DL: 60.6. With DL, Dom-spec yields significantly improved performance in 4/6 cases, compared to 2/6 with Pre-def (DL). This is apparently due to the penalization of relevant source-specific terms that have wider coverage compared to the pre-defined terms in Pre-def. Since the entropy-based attention regularization by Attanasio et al. (2022) do not use the target domain unlabeled instances for term-extraction, it may not be optimal for cross-domain settings. The large improvement with Pre-def (DL) for Vidgen \( \rightarrow \) Waseem (70.3) could be attributed to the fact that Vidgen involves a wide variety of identity terms. Thus, penalizing the pre-defined identity terms might result in higher emphasis on more generalizable hate-speech content. While only this particular case drives the high average performance with Pre-def (DL), Dom-spec

#### Table 2: Macro-F1 (±std-dev) on source \( \rightarrow \) target pairs. H : HatEval, V : Vidgen, W : Waseem. Bold denotes the best score and underline the second best in each column. * denotes statistically significant improvement compared to Van-MLM-FT with paired bootstrap (Dror et al., 2018; Efron and Tibshirani, 1993), 95% confidence interval.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>H ( \rightarrow ) V</th>
<th>V ( \rightarrow ) H</th>
<th>H ( \rightarrow ) W</th>
<th>W ( \rightarrow ) H</th>
<th>W ( \rightarrow ) V</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Per-MLM-FT</td>
<td>56.8±1.3</td>
<td>66.2±1.2</td>
<td>70.0±2.5</td>
<td>50.9±2.1</td>
<td>61.4±2.4</td>
<td>43.3±1.9</td>
</tr>
<tr>
<td>Liu and Avci (2019)</td>
<td>45.1±4.5</td>
<td>59.5±0.7</td>
<td>57.2±3.8</td>
<td>52.6±0.8</td>
<td>57.1±2.7</td>
<td>39.6±2.0</td>
</tr>
<tr>
<td>MLM + Kennedy et al. (2020) (a)</td>
<td>55.4±2.0</td>
<td>65.5±0.8</td>
<td>64.1±1.4</td>
<td>54.8±1.3</td>
<td>59.2±1.8</td>
<td>44.5±2.9</td>
</tr>
<tr>
<td>MLM + Kennedy et al. (2020) (b)</td>
<td>54.6±2.9</td>
<td>65.7±0.9</td>
<td>67.3±1.2</td>
<td>54.7±2.2</td>
<td>62.3±1.7</td>
<td>46.6±1.5</td>
</tr>
<tr>
<td>BERT PERL</td>
<td>54.1±0.7</td>
<td>60.0±0.6</td>
<td>60.1±2.0</td>
<td>55.2±0.7</td>
<td>55.5±1.0</td>
<td>37.8±1.2</td>
</tr>
<tr>
<td>BERT-AAD</td>
<td>56.6±1.3</td>
<td>53.9±3.5</td>
<td>68.8±2.5</td>
<td>50.7±1.4</td>
<td>48.3±4.7</td>
<td>53.0±1.7</td>
</tr>
<tr>
<td>HATN</td>
<td>48.4±1.6</td>
<td>59.1±0.4</td>
<td>59.7±2.9</td>
<td>51.4±1.8</td>
<td>60.0±2.6</td>
<td>45.4±2.7</td>
</tr>
<tr>
<td>MLM + Sarwar and Murdock (2022)</td>
<td>55.0±1.9</td>
<td>66.2±2.0</td>
<td>68.8±1.1</td>
<td>48.2±3.1</td>
<td>57.9±1.4</td>
<td>36.2±1.1</td>
</tr>
<tr>
<td>MLM + Attanasio et al. (2022)</td>
<td>54.9±1.6</td>
<td>65.6±1.4</td>
<td>64.1±5.0</td>
<td>52.4±3.7</td>
<td>62.5±8.0</td>
<td>43.5±2.3</td>
</tr>
<tr>
<td>MLM + Comb + ( \chi^2 )-test</td>
<td>57.9±1.6</td>
<td>67.4±1.5</td>
<td>71.3±1.0</td>
<td>48.9±4.0</td>
<td>60.0±2.0</td>
<td>46.5±2.9</td>
</tr>
<tr>
<td>Pre-def (DL)</td>
<td>58.9±0.7</td>
<td>66.3±0.7</td>
<td>70.7±1.8</td>
<td>52.3±3.0</td>
<td>60.8±2.2</td>
<td>46.9±2.5</td>
</tr>
<tr>
<td>Dom-spec (( \alpha \nabla \alpha ))</td>
<td>53.8±1.8</td>
<td>66.8±0.7</td>
<td>70.9±1.0</td>
<td>51.5±2.1</td>
<td>59.8±1.5</td>
<td>45.9±3.1</td>
</tr>
<tr>
<td>Comb (Dom-spec + Pre-def) (( \alpha \nabla \alpha ))</td>
<td>58.7±2.1</td>
<td>67.7±1.0</td>
<td>70.9±1.0</td>
<td>51.5±2.1</td>
<td>59.8±1.5</td>
<td>45.9±3.1</td>
</tr>
<tr>
<td>Pre-def (DL)</td>
<td>58.3±1.4</td>
<td>66.3±1.4</td>
<td>71.2±1.4</td>
<td>52.9±1.9</td>
<td>63.6±2.0</td>
<td>48.8±4.7</td>
</tr>
<tr>
<td>Dom-spec (DL)</td>
<td>58.8±0.6</td>
<td>66.4±1.2</td>
<td>71.3±0.9</td>
<td>51.1±2.2</td>
<td>62.5±2.2</td>
<td>46.6±1.9</td>
</tr>
<tr>
<td>Comb (Dom-spec + Pre-def) (DL)</td>
<td>58.4±1.4</td>
<td>66.7±1.0</td>
<td>71.3±0.9</td>
<td>51.1±2.2</td>
<td>62.5±2.2</td>
<td>46.6±1.9</td>
</tr>
</tbody>
</table>
penalization of source-specific terms, the source domain classifier learns to focus on the wider context of the instances. For example, we observe that in the case of Waseem → HatEval, the automatically extracted terms related to the role of women in sports, such as {sports, sexist, gaming, football, commentary, competition, ...}. Note that Wiegand et al. (2019) also mention that these terms cause domain or topic bias in Waseem, restricting generalizability. See Appendix C for more examples.

5 Conclusion

We proposed a DA approach for automatic extraction and penalization of source domain-specific terms that have higher attributions towards the hate-speech labels, to improve cross-domain hate-speech detection. The results demonstrated consistent improvements on the target domain. These results should motivate further research on domain adaptation in hate-speech and building classifiers that can generalize well to the concept of hate. Finally, it would be interesting in applying our approach to other tasks such as rumor and misinformation detection (Mu and Aletras, 2020; Mu et al., 2022).

Ethical Considerations

This work serves as a means to build more robust hate-speech detection models that can make proper use of the existing curated hate-speech resources and adapt well on new resources or social-media comments, which have not been well-annotated due to time and cost constraints. The hate-speech resources used for the work are publicly available and cited appropriately, wherein the authors have discussed the sampling techniques and annotation guidelines in detail. The hate-speech examples presented in the paper are only intended for research purposes and better analysis of the models explored. The terms extracted and penalized in this work are not meant to be used off-the-shelf, but the approach should serve as a starting point for research on model-debugging and building more generalizable hate-speech classifiers.

Acknowledgments

This work was supported partly by the french PIA project “Lorraine Université d’Excellence”, reference ANR-15-IDEX-04-LUE. Experiments presented in this article were carried out using the
Grid’5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). We thank the anonymous reviewers for their valuable feedback and suggestions.

References


George Chrysostomou and Nikolaos Aletras. 2022b. Flexible instance-specific rationalization of NLP models. AAAI.


Fabio Polletto, Valerio Basile, Cristina Bosco, Viviana Patti, and Marco Antonio Stranisci. 2019. Annotating hate speech: Three schemes at comparison. In CLiC-IT.


### Differences across Datasets

The datasets HatEval (Basile et al., 2019) and Waseem (Waseem and Hovy, 2016) have been sampled from Twitter. HatEval has primarily been collected in the year 2018 using a combination of sampling strategies, including keyword-based sampling (with both neutral and derogatory words), collecting the history of identified perpetrators and monitoring the potential victims of hate. It mainly consists of hate against women and immigrants. In the case of Waseem, tweets are collected particularly using keyword-based sampling in or before 2016, with keywords that are likely to co-occur with hateful content. Wiegand et al. (2019) discuss the presence of a large amount of topic-bias in the dataset Waseem. Since this dataset is available as tweet-IDs, we observe that in the crawled dataset, many tweets flagged as racist are missing, and have most likely been deleted already. Thus, the majority of available hateful content in this dataset is directed against women. The topics discussed in these two datasets are also quite different.
Vidgen\(^5\) (Vidgen et al., 2021), on the other hand, is a dataset generated using a human and model-in-loop process. This process results in adding several perturbations and instances, which are difficult to classify, aimed at making the dataset robust. Besides, it consists of hateful content directed against a wide array of target groups, e.g. black, gay, muslim, disabled, etc., along with different forms of hate such as derogation, threatening language, animosity, support for hateful entities and dehumanization. Thus there is a substantial amount of differences across these datasets in terms of collection time-frames, sampling strategies, targets of hate, forms of hate, vocabulary used and the like.

For pre-processing the datasets, we remove the URLs, split the hashtags using CrazyTokenizer\(^6\) and lowercase the terms.

B Implementation Details and Hyper-parameter Tuning

We use the pre-trained BERT-base (Devlin et al., 2019) uncased model\(^7\) (Wolf et al., 2020) for our experiments. We run both the Masked Language Model (MLM) training on the unlabeled target domain training data \(D_T^{\text{train}}\), and the subsequent supervised fine-tuning on the source domain training data \(D_S^{\text{train}}\) for 6 epochs with a batch size of 8 for all the BERT baselines and Dom-spec. We use the AdamW optimizer with decoupled weight decay regularization (Loshchilov and Hutter, 2019), having a weight decay of \(10^{-4}\). We use a learning rate of \(3 \times 10^{-5}\) for the MLM training and \(1 \times 10^{-5}\) for the supervised fine-tuning, with the epsilon parameter set to \(1 \times 10^{-8}\).

We use the original implementations provided by the respective authors of all the baselines except for Sarwar and Murdock (2022). We implement the data-augmentation approach by Sarwar and Murdock (2022) ourselves, as there is no available implementation. We follow the description provided in the paper and label all the terms in the hateful instances of the source domain that have a match with hatebase.org\(^8\) for training a sequence tagger. However, while finding the matches, we do not tokenize the multi-word phrases in hatebase.org. We lowercase the terms from hatebase.org and look for an exact match of a term in the source domain.

For Pre-Def, we combine the curated list of identity terms provided by Liu and Avci (2019) and Kennedy et al. (2020) and penalize their attribution scores. We perform hyper-parameter tuning and model selection with early-stopping on a small amount of labeled target domain validation set \(D_T^{\text{val}}\) using the macro-F1 score for the proposed approach as well as for all the baselines. The hyper-parameter \(\lambda\), both for the proposed approach and Pre-Def, is selected from the range \(\lambda \in \{0.01, 0.05, 0.1, 1.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0\}\), using a random seed by tuning over \(D_T^{\text{val}}\). We set the value of \(M\) to \(250\) and \(N\) to \(750\) for all our experiments.

C Terms Extracted

The full-list of penalized terms (BERT WordPieces) \(\text{te}\(^5\) across epochs for the examples listed in Section 4.2, is given below.

Waseem →HatEval

- **Epoch 1:** \{college, sports, feminism, la, magnetic, used, unique, ##ava, speech, ##js, tr, ##cking, object, chu, result, ki, bus, ##is, adopt, referring, ##roids, handed, ##em, sh, ##omp, unconscious, anger, gamer, prove, xbox, tri, skill, judgment, tool, block, single, harassment, size, georgia, involved, ##ism, studying, voices, possible, gaming, pl, ##il, helped, ##ke, survey, equality\}

- **Epoch 2:** \{feminism, used, football, awesome, equal, ##cking, object, ##ification, interest, feminist, ##rra, scientist, ##al, ignorance, bodies, ##work, later, ##nk, troll, ##ss, based, adopt, ##icing, quality, sister, unconscious, criticisms, pro, notch, xbox, tri, unfair, rap, meanwhile, impression, single, harassment, bonus, georgia, constant, sex, ##ist, possible, click, competition, ##per, swedish, ##eral, november, write, eventually, equality\}

- **Epoch 3:** \{sham, anger, pull, used, focus, speech, ashley, object, interest, bringing, #na, eye, ##nk, later, quality, ##roids, oppressive, rain, ##omp, statistics, nsw, content, notch, museum, unconscious, typically, tri, ##ol, unfair, writing, ##chan, georgia, constant, annie, ra, weights, click, ##il, furniture, helped, shopping, football, commentary, equality\}

- **Epoch 4:** \{minded, kat, used, equal, focus, ##hand, tr, ##cking, chu, interest, bringing,
Table 4: Effect of domain adaptation for the target on the source domain performance; Source-domain macro average F1 scores (mean±std-dev) are obtained after MLM training on the unlabeled target domain and penalizing the source specific terms while adapting the model to the target domain (present at the right hand side of the arrows) using Dom-spec. H : HatEval, V : Vidgen, W : Waseem. Van-FT: BERT model evaluated in-domain without MLM training on the target domain.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>HatEval</th>
<th>Vidgen</th>
<th>Waseem</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Van-FT</td>
<td>43.3±1.8</td>
<td>85.1±0.5</td>
<td>85.4±0.7</td>
<td>71.3</td>
</tr>
</tbody>
</table>

Performance on source domain (left of arrows) while applying domain adaptation for the target (right of arrows)

<table>
<thead>
<tr>
<th></th>
<th>H → V</th>
<th>H → W</th>
<th>V → H</th>
<th>V → W</th>
<th>W → H</th>
<th>W → V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dom-spec (αVα)</td>
<td>42.4±2.5</td>
<td>42.0±2.1</td>
<td>84.0±0.9</td>
<td>85.5±1.0</td>
<td>85.1±0.7</td>
<td>83.8±0.8</td>
</tr>
<tr>
<td>Dom-spec (DL)</td>
<td>41.7±3.7</td>
<td>40.5±4.4</td>
<td>83.9±0.7</td>
<td>82.6±1.5</td>
<td>84.7±1.2</td>
<td>81.1±2.7</td>
</tr>
</tbody>
</table>

Few of the extracted terms get repeated in subsequent epochs as a single epoch may not be sufficient to reduce the effect of a term and it may appear in the next epoch as well. Moreover, as the training progresses, the model may learn new patterns, and some extracted terms may reappear and disappear again due to the penalization.

Following is a non-hateful example in HatEval, wrongly classified by Van-MLM-FT but correctly classified by Dom-spec (The darker the shades, the higher the attribution scores assigned):

**Van-MLM-FT:** Unfortunately you are in a sticky size my only problem is replacing my shoes has been a b*tch

**Dom-spec (DL):** Unfortunately you are in a sticky size my only problem is replacing my shoes has been a b*tch

Following is a non-hateful example in Waseem, wrongly classified by Van-MLM-FT, but correctly classified by the proposed approach (The darker the shades, the higher the attribution scores assigned):

**Van-MLM-FT:** Omg I am lisening to an apple genius dude tell this old woman how to use email and it is adorable

**Dom-spec (DL):** Omg I am listening to an apple genius dude tell this old woman how to use email and it is adorable

D In-domain Performance

Table 4 presents, as a reference, the in-domain macro-F1 scores using BERT supervised fine-tuning (Van-FT) without the MLM training on the target domain. In this case, the model is tuned over
the in-domain validation set. The HatEval dataset is part of a shared task and involves a challenging test set with low in-domain performance. Table 4 displays the source-domain scores obtained when source-specific terms are penalized, while adapting to the target domain using Dom-spec, where the model is tuned over the target domain validation set. The drop in in-domain performance is expected as Dom-spec is aimed at making the model best-suited to the target domain. However, the overall performance with Dom-spec is comparable to that of BERT Van-FT.

### E List of Identity Terms for Pre-Def

The combined list of pre-defined curated identity terms from Liu and Avci (2019) and Kennedy et al. (2020) are given below:

\{lesbian, gay, bisexual, trans, cis, queer, lgbt, lgbtq, straight, heterosexual, male, female, non-binary, african, african american, european, hispanic, latino, latina, latinx, canadian, american, asian, indian middle eastern, chinese, japanese, christian, buddhist, catholic, protestant, sikh, taoist, old, older, young, younger, teenage, millenial, middle aged, elderly, blind, deaf, paralyzed, muslim, jew, jews, white, islam, blacks, muslims, women, whites, gay, black, democrat, islamc, allah, jewish, lesbian, transgender, race, brown, woman, mexican, religion, homosexual, homosexuality, africans\}

### F Computational Efficiency

The per-epoch training time for Dom-spec, while performing adaptation of different source domain models, are presented in Table 5. Dom-spec ($\alpha \nabla \alpha$) takes less than double the time taken by Van-MLM-FT to train, and Dom-spec (DL) takes roughly 4.5 times of the training time taken by Van-MLM-FT.

<table>
<thead>
<tr>
<th>Approaches</th>
<th>HatEval</th>
<th>Vidgen</th>
<th>Waseem</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Van-MLM-FT</td>
<td>1 m 20 s</td>
<td>3 m 49 s</td>
<td>2 m 10 s</td>
</tr>
<tr>
<td>Dom-spec ($\alpha \nabla \alpha$)</td>
<td>2 m 30 s</td>
<td>7 m</td>
<td>3 m 17 s</td>
</tr>
<tr>
<td>Dom-spec (DL)</td>
<td>4 m</td>
<td>18 m</td>
<td>8 m 16 s</td>
</tr>
</tbody>
</table>

Table 5: Per epoch training time on different source domains.