Transferring Knowledge via Neighborhood-Aware Optimal Transport for Low-Resource Hate Speech Detection

Tulika Bose Irina Illina Dominique Fohr
Universite de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{tulika.bose, illina, dominique.fohr}@loria.fr

Abstract

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

The concerning rise of hateful content on online platforms has increased the attention towards automatic hate speech detection, commonly formulated as a supervised classification task. State-of-the-art deep learning-based approaches usually require a substantial amount of labeled resources for training. However, annotating hate speech resources is expensive, time-consuming, and often harmful to the annotators. This creates a pressing need to transfer knowledge from the existing labeled resources to low-resource hate speech corpora with the goal of improving system performance. For this, neighborhood-based frameworks have been shown to be effective. However, they have limited flexibility. In our paper, we propose a novel training strategy that allows flexible modeling of the relative proximity of neighbors retrieved from a resource-rich corpus to learn the amount of transfer. In particular, we incorporate neighborhood information with Optimal Transport, which permits exploiting the geometry of the data embedding space. By aligning the joint embedding and label distributions of neighbors, we demonstrate substantial improvements over strong baselines, in low-resource scenarios, on different publicly available hate speech corpora.

1 Introduction

With the alarming spread of Hate Speech (HS) in social media, Natural language Processing techniques have been used to develop automatic HS detection systems, typically to aid manual content moderation. Although deep learning-based approaches (Mozafari et al., 2019; Badjatiya et al., 2017) have become state-of-the-art in this task, their performance depends on the size of the labeled resources available for training (Lee et al., 2018; Alwosheel et al., 2018).

Annotating a large corpus for HS is considerably time-consuming, expensive, and harmful to human annotators (Schmidt and Wiegand, 2017; Malmasi and Zampieri, 2018; Poletto et al., 2019; Sarwar et al., 2022). Moreover, models trained on existing labeled HS corpora have shown poor generalization when evaluated on new HS content (Yin and Zubiaga, 2021; Arango et al., 2019; Swamy et al., 2019; Karan and Šnajder, 2018). This is due to the differences across these corpora, such as sampling strategies (Wiegand et al., 2019), varied topics of discussion (Florio et al., 2020; Saha and Sindhwani, 2012), varied vocabularies, and different victims of hate. Thus, to address these challenges, here we aim to devise a strategy that can effectively transfer knowledge from a resource-rich source corpus with a higher amount of annotated content to a low-resource target corpus with fewer labeled instances.

One popular way to address this is transfer learning. For instance, Mozafari et al. (2019) fine-tune a large-scale pre-trained language model, BERT (Devlin et al., 2019), on the limited training examples in HS corpora. Further, a sequential transfer, following Garg et al. (2020), can be performed where a pre-trained model is first fine-tuned on a resource-rich source corpus and subsequently fine-tuned on the low-resource target corpus. Since this may risk forgetting knowledge from the source, the source and target corpora can be mixed for training (Shnarch et al., 2018). Besides, to learn target-specific patterns without forgetting the source knowledge, Meftah et al. (2021) augment pre-trained neurons from the source model with randomly initialized units for transferring knowledge to low-resource domains.

Recently, Sarwar et al. (2022) argue that traditional transfer learning strategies are not systematic. Therefore, they model the relationship between a source and a target corpus with a neighborhood framework and show its effectiveness in transfer learning for content flagging. They model the in-

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 453–467
November 20–23, 2022. ©2022 Association for Computational Linguistics

teraction between a query instance from the target and its neighbors retrieved from the source. This interaction is modeled based on their label agreement – whether the query and its neighbors have the same labels – while using a fixed neighborhood size. However, different neighbors may have varying levels of proximity to the queried instance based on their pair-wise cosine similarities in a sentence embedding space. Therefore, intuitively, the neighbors should also be weighted according to these similarity scores.

We hypothesize that simultaneously modeling the pair-wise distances between instances from the low-resource target and their respective neighbors from the resource-rich source, along with their label distributions should result in a more flexible and effective transfer. With this aim, we propose a novel training strategy where the model learns to assign varying importance to the neighbors corresponding to different target instances by optimizing the amount of pair-wise transfer. This transfer is learned without changing the underlying model architecture. Such optimization can be efficiently performed using *Optimal Transport* (OT) (Peyré and Cuturi, 2019; Villani, 2009; Kantorovich, 2006) due to its ability to find correspondences between instances while exploiting the underlying geometry of the embedding space. Our contributions are summarised as follows:

- We address HS detection in low-resource scenarios with a flexible and systematic transfer learning strategy.
- We propose novel incorporation of neighborhood information with joint distribution Optimal Transport. This enables learning of the amount of transfer between pairs of source and target instances considering both (i) the similarity scores of the neighbors and (ii) their associated labels. To the best of our knowledge, this is the first work that introduces Optimal Transport for HS detection.
- We demonstrate the effectiveness of our approach through considerable improvements over strong baselines, along with quantitative and qualitative analysis on different HS corpora from varied platforms.

2 Related Works

2.1 Hate Speech Detection

Deep Neural Networks, especially the transformer-based models, such as the pre-trained BERT, have dominated the field of HS detection in the past few years (Alatawi et al., 2021; D'Sa et al., 2020; Glavaš et al., 2020; Mozafari et al., 2019).

Wiegand et al. (2019); Arango et al. (2019) raise concerns about data bias present in most HS corpora, which results in overestimated within-corpus performance. They, therefore, recommend cross-corpus evaluations as more realistic settings. Bigoulaeva et al. (2021); Bose et al. (2021); Pamungkas et al. (2021) perform such cross-corpus evaluations in this task with no access to labeled instances from the target. However, Yin and Zubiaga (2021); Wiegand et al. (2019) report fluctuating or degraded performance across corpora. As pointed out by Sarwar et al. (2022), in real-life scenarios, most online platforms could invest in obtaining at least some labeled training instances for deploying an HS detection system. Thus, we study a more realistic setting where a limited amount of labeled content is available in the target corpus.

2.2 Neighborhood Framework

k-Nearest Neighbors (kNN)-based approaches have been successfully used in the literature for an array of tasks such as language modeling (Khandelwal et al., 2020), question answering (Kassner and Schütze, 2020), dialogue generation (Fan et al., 2021), etc. Besides, kNN classifiers have been used for HS detection (Prasetyo and Samudra, 2022; Briliani et al., 2019), which typically predict the class of an input instance through a simple majority voting using its neighbors in the training data.

Recently, Sarwar et al. (2022) propose a neighborhood framework kNN$^+$ for transfer learning in cross-lingual low-resource settings. They show that a simple kNN classifier is prone to prediction errors as the neighbors may have similar meanings, but opposite labels. They, instead, model the interactions between the target corpus instances, treated as queries, and their nearest neighbors retrieved from the source. This neighborhood interaction is modeled based on whether a query and its neighbors have the same or different labels. In their best performing framework (in cross-lingual setting) of Cross-Encoder kNN$^+$, Sarwar et al. (2022) obtain representations of concatenated query-neighbor pairs to learn such neighborhood interactions.

However, Sarwar et al. (2022) do not consider *the varying levels of the proximity of different neighbors to the query.* Besides, a mini-batch in their framework comprises a query and all its neighbors. For fine-tuning large language models like BERT, the batch size needs to be kept small due to resource constraints. This could limit the neighborhood size in their framework. This is different from our approach, where the neighborhood size is scalable.

2.3 Optimal Transport

Optimal Transport (OT) has become increasingly popular in diverse NLP applications, as it allows comparing probability distributions in a geometrically sound manner. These include machine translation (Xu et al., 2021), interpretable semantic similarity (Lee et al., 2022), rationalizing text matching (Swanson et al., 2020), etc. Moreover, OT has been successfully used for domain adaptation in audio, images, and text (Olvera et al., 2021; Damodaran et al., 2018; Chen et al., 2020). In this work, we perform novel incorporation of nearest neighborhood information with OT. Besides, to the best of our knowledge, this is the first work that introduces OT to the HS detection task.

3 Proposed Approach

Our problem setting involves a low-resource target corpus X^t with a limited amount of labeled training data $(X^t_{train}, Y^t_{train}) = \{x^t_i, y^t_i\}_{i=1}^{n_t}$ and a resource-rich source corpus X^s from a different distribution with a large number of annotated data $(X^s_{train}, Y^s_{train}) = \{x^s_i, y^s_i\}_{i=1}^{n_s}$. Given such a setting, we hypothesize that transferring knowledge from the nearest neighbors in the source should improve the performance on the insufficiently labeled target. Furthermore, to provide additional control to the model, we propose a systematic transfer. With this transfer mechanism, a model can *learn* different weights assigned to the neighbors in X^s_{train} based on their proximity to the instances in X^t_{train} simultaneously in a sentence embedding space and the label space. For this, we incorporate neighborhood information with Optimal Transport (OT), as OT can learn correspondences between instances from X^s_{train} and X^t_{train} by exploiting the underlying embedding space geometry.

3.1 Joint Distribution Optimal Transport

In this work, we use the joint distribution optimal transport (JDOT) framework (Courty et al., 2017) following the works of Damodaran et al. (2018); Fatras et al. (2021), proposed for unsupervised domain adaptation in deep embedding spaces. The framework aligns the joint distribution $P(Z, Y)$ of the source and the target domains, where Z is the embedding space through a mapping function $g(.)$, and Y is the label space. For a discrete setting, let $\mu_s = \sum_i^{n_s} a_i\, \delta_{g(x^s_i), y^s_i}$ and $\mu_t = \sum_i^{n_t} b_i\, \delta_{g(x^t_i), y^t_i}$ be two empirical distributions on the product space of $Z \times Y$. Here $\delta_{g(x_i), y_i}$ is the Dirac function at the position $(g(x_i), y_i)$, and a_i, b_i are uniform probability weights, i.e. $\sum_i^{n_s} a_i = \sum_i^{n_t} b_i = 1$.

The 'balanced' OT problem (OT_b), as defined by Kantorovich (2006), seeks for a transport plan γ in the space of the joint probability distribution $\Pi(\mu_s, \mu_t)$, with marginals μ_s and μ_t, that minimizes the cost of transport from μ_s to μ_t, as:

$$\begin{aligned} OT_b(\mu_s, \mu_t) = \min_{\gamma \in \Pi(\mu_s, \mu_t)} & \sum_{i,j} \gamma_{i,j} c_{i,j} \\ s.t. \quad & \gamma \mathbf{1}_{n_t} = \mu_s, \gamma^T \mathbf{1}_{n_s} = \mu_t \end{aligned} \tag{1}$$

Here $c_{i,j}$ is an entry in a cost matrix $C \in R^{n_s \times n_t}$, representing the pair-wise cost (see Section 3.2), and $\mathbf{1}_n$ is a vector of ones with dimension n. Each entry $\gamma_{i,j}$ indicates the amount of transfer from location i in the source to j in the target.

The constraint on γ requires that all mass from μ_s is transported to μ_t. However, this can be alleviated through relaxation, leading to the 'unbalanced' OT (OT_u) (Benamou, 2003), as:

$$\begin{aligned} OT_u(\mu_s, \mu_t) = \min_{\gamma \in \Pi(\mu_s, \mu_t)} & \sum_{i,j} \gamma_{i,j} c_{i,j} + \Lambda\,; \\ \text{where} \quad & \Lambda = \epsilon\, \Omega(\gamma) + \lambda \left(\mathrm{KL}(\gamma \mathbf{1}_{n_t}, \mu_s) + \mathrm{KL}(\gamma^T \mathbf{1}_{n_s}, \mu_t) \right) \\ s.t. \quad & \gamma \geq 0 \end{aligned} \tag{2}$$

KL is the Kullback-Leibler divergence that allows the relaxation of the marginal constraint on γ. λ is the marginal relaxation coefficient. $\Omega(\gamma) = \sum_{i,j} \gamma_{i,j} log(\gamma_{i,j})$ corresponds to the entropic regularization term, which allows fast computation of the OT distances (Cuturi, 2013). ϵ is the entropy coefficient.

For models with a high-dimensional embedding space like ours, Fatras et al. (2021) propose to make the computation of OT losses scalable using the mini-batch OT. Thus, for every mini-batch, we sample an equal number of instances, given by the batch size m, from X^s_{train} and X^t_{train}, which makes $C \in R^{m \times m}$ and γ square matrices. As discussed by Fatras et al. (2021), since the transport plan at the mini-batch level is much less sparse, it may result in undesired pairings between instances if computed by Equation 1. To counteract this effect, we rely on the more robust version of OT as formulated in Equation 2. Thus, we adopt the *joint distribution entropy regularized unbalanced mini-batch OT* for our framework, henceforth simply referred to as OT. Note that this framework does not modify the underlying model architecture used for classification, but only introduces a new training strategy.

3.2 Neighborhood-aware OT (OT^{NN})

In the above joint distribution framework, the cost matrix C is expressed as the weighted combination of the costs in the embedding and the label spaces:

$$c_{i,j}(g(x_i^s), y_i^s; g(x_j^t), y_j^t) = \alpha\, d(g(x_i^s), g(x_j^t)) + \beta\, L(y_i^s, y_j^t) \tag{3}$$

$d(.,.)$ denotes the *embedding distance* (ED), which is a squared l_2 distance between the corresponding embeddings. $L(.,.)$ is *label-consistency loss* (LC), which is a cross-entropy loss that enforces a match between the label of the i^{th} source instance and that of the j^{th} target instance. α and β are scalar values. Minimizing the cost in Equation 3 results in aligning instances from the source and the target that simultaneously share similar representations and common labels.

We adapt C to account for k nearest neighbors of the target instances in X_{train}^t from the source X_{train}^s. Since BERT is not optimal for semantic similarity search (Reimers and Gurevych, 2019), we extract the neighbors using the Sentence-BERT (SBERT) model (Reimers and Gurevych, 2019). SBERT provides sentence embeddings that can be easily compared using cosine similarity. We hypothesize that allowing transfers to occur only from the corresponding neighbors in the source to the target should result in more effective learning.

For this, we explicitly assign the value $\max(C)$ to $c_{i,j}$ in C whenever the i^{th} source and j^{th} target instances are not neighbors, considering the nearest neighborhood space of k neighbors. Besides, we use the SBERT distances as the embedding distance in Equation 3. This distance, in addition to the label consistency term, ensures that γ is learned to allow a higher amount of transfer from neighbors in X_{train}^s that are simultaneously (i) closer in the SBERT space and (ii) share the same label with an instance in X_{train}^t, compared to the neighbors that are further away and/or have opposite labels.

Note that even though we use a neighborhood size of k, the target instances do not attend equally to all of their k neighbors. This is because if the distance between a target instance x_j^t and its top n^{th} neighbor (x_i^s) from the source, within the neighborhood size of k (i.e. $n < k$) is comparatively large, their corresponding (i, j)-th entry in C would have a larger value. This would comparatively reduce the transfer *even if they share common labels*. Thus, for a neighbor with the same label as the target instance, the higher its SBERT distance from the target instance, the lower the amount of transfer. This results in more flexibility where the model can learn from the relevant neighbors corresponding to every target instance.

In addition to the OT loss from Equation 2, we introduce the cross-entropy losses for the training instances from both X_{train}^t and X_{train}^s in the final loss function, as required by our classification task. Our final loss function is given by Equation 4. Here $g(.)$ encodes a given input using the pre-trained BERT encoder to the BERT embedding space by extracting the fine-tuned [CLS] token representation of the last hidden layer. $f(.)$ denotes the classifier, which is one fully connected layer. θ_s and θ_t are the weights assigned to the source and the target cross-entropy losses, respectively.

$$\begin{aligned}\mathrm{OT}^{NN} = \min_{\gamma,f,g} \quad & \theta_s \frac{1}{m}\sum_i L_s\left(y_i^s, f(g(x_i^s))\right) + \sum_{i,j}\gamma_{i,j}c_{i,j} \\ & + \Lambda + \theta_t \frac{1}{m}\sum_j L_t\left(y_j^t, f(g(x_j^t))\right)\end{aligned} \tag{4}$$

Solving the optimization problem: Following Damodaran et al. (2018), we adopt a two-step procedure to solve the above optimization problem at the mini-batch level. We first compute the optimal γ by fixing the model parameters of f and g.

$$\min_{\gamma} \sum_{i,j}\gamma_{i,j}\left(\alpha\, d(g_{sbert}(x_i^s), g_{sbert}(x_j^t)) + \beta\, L(y_i^s, y_j^t)\right) + \Lambda \tag{5}$$

We use the SBERT embeddings through the mapping function $g_{sbert}(.)$ here instead of the learned BERT embeddings to compute the ED loss. This is done so that the γ is updated based on the semantic proximity in the SBERT space. y_i^s and y_j^t are the ground truth labels for the instances x_i^s and x_j^t from X_{train}^s and X_{train}^t, respectively. In the next step, the model parameters of f and g are learned while fixing γ obtained from Equation 5, denoted as $\hat{\gamma}$.

$$\min_{f,g} \quad \sum_{i,j} \hat{\gamma}_{i,j} \left(\alpha \, d(g(x_i^s), g(x_j^t)) + \beta \, L(f(g(x_i^s)), y_j^t)\right) \\ + \theta_s \frac{1}{m} \sum_i L_s \left(y_i^s, f(g(x_i^s))\right) + \theta_t \frac{1}{m} \sum_j L_t \left(y_j^t, f(g(x_j^t))\right) \tag{6}$$

The first part of Equation 6 allows the model to learn from the instances in X^s_{train} that are consistent in terms of both the embedding space (ED loss) and the label space (LC loss) with the instances in X^t_{train}. Here we use $g(.)$, instead of $g_{sbert}(.)$, to compute ED so that g learns from the SBERT space through $\hat{\gamma}$. For the LC loss, we use the predicted labels for x_i^s from the source and the actual labels y_j^t corresponding to x_j^t from the target. This is done to update the model parameters of f and g based on the target labels and bring source instances that have common labels closer to the target instances. We have provided an illustration of the training strategy of OT^{NN} in Figure 3 of Appendix A.

We propose different variants of OT^{NN}:

OT^{NN}: In this variant, we do not use the source cross-entropy loss term in Equation 4, thus effectively having $\theta_s = 0$.

OT$^{NN}_{\text{pre-select}}$: Prior to the training, we pre-select the k nearest neighbors from X^s_{train} corresponding to every instance in X^t_{train}, instead of training with all the source instances. Here also $\theta_s = 0$.

OT^{NN} + sloss: This is OT^{NN} with source cross-entropy loss (sloss), thus having $\theta_s = 1$.

OT$^{NN}_{\text{pre-select}}$ + sloss: This is similar to the second variant, with $\theta_s = 1$. Here, sloss is computed only on the pre-selected source instances.

4 Experimental Settings

4.1 Corpus Description

We perform experiments with three standard HS corpora, namely, *Waseem* (Waseem and Hovy, 2016), *Vidgen* (Vidgen et al., 2021), and *Ethos* (Mollas et al., 2022), as they are collected using different sampling strategies across varied platforms. Following Wiegand et al. (2019); Swamy et al. (2019), we use the labels of *hate* and *non-hate*, where the former involves all forms of hate.

Waseem is a Twitter corpus comprising hate against women and ethnic minorities. We obtain 10.9K tweets in total from the tweet IDs, of which 26.8% instances belong to the *hate* class. *Vidgen* is collected using a human-and-model-in-the-loop process aimed at making the corpus robust. It covers hate against diverse social groups, like blacks, women, muslims, immigrants, etc. with a total of 41144 instances, of which 53.9% is labeled as *hate*. *Ethos* comprises 998 instances from YouTube and Reddit, of which 43.4% are *hate* instances. Even with fewer instances, it is made diverse with an active learning-based sampling strategy, ensuring a balance with respect to different hateful aspects. See Appendix B for further details on the corpora.

For our experiments, we create two different versions of every corpus depending on its use as the source or the target, as presented in Table 1.

Corpus	Number of comments		
Source setting			
	Train		
Waseem$_{\text{src}}$	8720		
Vidgen$_{\text{src}}$	32924		
Ethos$_{\text{src}}$	998		
Target setting			
	Train	**Validation**	**Test**
Waseem$_{\text{tar}}$	400	100	1090
Vidgen$_{\text{tar}}$	400	100	4120
Ethos$_{\text{tar}}$	400	100	200

Table 1: Corpus statistics.

Source setting: In the absence of available standard splits, we randomly sample 80% of *Waseem* as the train set, resulting in 8720 instances. For *Vidgen*, we use the original corpus-provided train split of 32924 instances. Since *Ethos* has a relatively small size, we use the entire corpus for training, when used as the source. We call the source versions of these corpora as *Waseem*$_{\text{src}}$, *Vidgen*$_{\text{src}}$ and *Ethos*$_{\text{src}}$. Note that the source corpus is only used for training, while its validation set is not used for our experiments. Instead, we use the corresponding validation and test sets of the low-resource target corpus.

Target setting: In order to simulate a low-resource scenario for the target, we down-sample the original training instances of the corpora to 500 instances. This yields three low-resource target corpora, namely, *Waseem*$_{\text{tar}}$, *Vidgen*$_{\text{tar}}$ and *Ethos*$_{\text{tar}}$. Furthermore, we split each of them in the 80-20 ratio to obtain their respective low-resource train (400) and validation (100) sets. For the test set from *Waseem*$_{\text{tar}}$, we sample 10% of the original data, disjoint from the train and validation sets, given by 1090 instances. We use the original test split of 4120 instances for *Vidgen*$_{\text{tar}}$. For *Ethos*$_{\text{tar}}$, we randomly sample 20% of the data, disjoint from the previous set of 500 instances, as the test set.

4.2 Baselines

We compare our approach with the following baseline approaches:

Target-FT: We fine-tune the pre-trained BERT on the train set of the low-resource target corpus.

Seq-FT: Here, we sequentially fine-tune the BERT model first on the resource-rich source corpus and then on the low-resource target corpus.

Mixed-FT: Here, we fine-tune BERT on a mix of the source and target corpora. Since the target instances are limited, we first over-sample them. Then, for every mini-batch of size m, we randomly sample m training instances each from the source and the target. We then combine their cross-entropy losses for updating the model parameters, as:

$$\min_{f,g} \quad \theta_s \tfrac{1}{m}\sum_i L_s(y_i^s, f(g(x_i^s))) + \theta_t \tfrac{1}{m}\sum_j L_t(y_j^t, f(g(x_j^t))) \quad (7)$$

This is similar to Equation 4 without the OT^{NN} losses.

***k*NN-FT:** For every target instance, we retrieve top-*k* neighbors from the source, ranked with cosine similarities over SBERT embeddings. This yields a subset of source instances that are neighbors to the target instances. We then fine-tune the BERT model with the strategy used for Mixed-FT.

***k*NN ranking:** Here, we predict the labels of the target instances using a majority voting strategy. This voting is done over the labels associated with the top-*k* retrieved neighbors from the source based on their cosine similarities.

Weighted *k*NN: This uses a weighted voting of the top-*k* neighbors. Here we compute the sum of cosine similarities of neighbors associated with every class. The class with the highest score is returned as the predicted label of the target instance.

CE *k*NN$^+$ + SRC: This is the Cross-Encoder-based neighborhood framework *k*NN$^+$, proposed by Sarwar et al. (2022), as discussed in Section 2.2. For a fair comparison, we use the pre-trained BERT as the base representation. We first train CE *k*NN$^+$ on the source (SRC) and then with the target instances and their neighbors from the source.

PretRand: This is a transfer learning strategy proposed by Meftah et al. (2021) for low-resource domain adaptation. They jointly learn a pre-trained branch in the target model with a normalized, weighted, and randomly initialized branch. This is done so that the model can learn target-specific patterns while retaining the source knowledge. For a fair comparison, we use the pre-trained BERT as the base model, which is first fine-tuned on the source. For the random branch, following the approach, we add a BiLSTM layer and a Fully Connected layer over the final hidden layer from BERT. The final predictions are obtained using an element-wise sum of the predictions from the two branches.

OT: Finally, we use OT to transfer knowledge from the source to the target using both the ED and LC losses, similar to Equation 4. However, this is done *without* incorporating any neighborhood information in both the cost matrix and the computation of γ.

4.3 Hyper-parameters

We train all the models for 10 epochs initialized with the pre-trained BERT-base (Devlin et al., 2019) uncased model (Wolf et al., 2020), with a maximum sequence length of 128 tokens. We use the Adam optimizer with a learning rate of 5×10^{-5}. Besides, we perform hyper-parameter tuning for k and model selection using the best F1 scores over the respective target corpus validation sets. After the preliminary experiments, we set α = 0.05, β = 10, ϵ = 0.2, λ = 0.5, and θ_t = 10 for all our experiments. We use a batch size of 32 for the OT^{NN} and the baselines, except CE-*k*NN$^+$. The latter inherently requires the batch size to be equal to the neighborhood size, as it provides query-neighborhood pairs as inputs to the model. See Appendix D for further details on the hyper-parameter tuning.

5 Results

5.1 Discussion

Table 2 shows the performance obtained with the baselines and the OT^{NN} variants across the test sets of three low-resource target corpora using different resource-rich source corpora. We also present the performance with Target-FT for reference. Following the prior work on HS detection (Sarwar et al., 2022; Attanasio et al., 2022), we use the F1 score of the hate class to report the performance, with an average F1 computed over five runs of the same experiments with different random initializations.

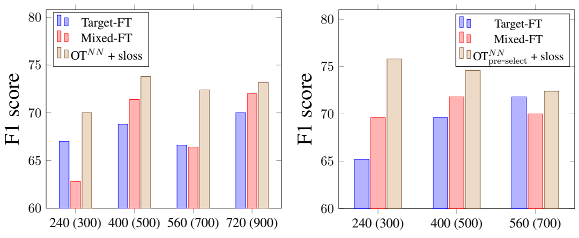
The results show that transferring knowledge from a resource-rich corpus to a low-resource corpus is generally helpful. The best scores in the

Target corpus	**Waseem$_{\mathtt{tar}}$**		**Vidgen$_{\mathtt{tar}}$**		**Ethos$_{\mathtt{tar}}$**	
Target-FT	64.0±2.1		68.8±3.2		69.6±6.4	
Source corpus	**Vidgen$_{\mathtt{src}}$**	**Ethos$_{\mathtt{src}}$**	**Waseem$_{\mathtt{src}}$**	**Ethos$_{\mathtt{src}}$**	**Vidgen$_{\mathtt{src}}$**	**Waseem$_{\mathtt{src}}$**
Seq-FT	63.2±2.1	65.0±1.1	67.0±2.2	70.8±3.9	**79.8**±0.7	70.2±3.1
Mixed-FT	61.2±2.7	66.6±2.2	69.8*±1.6	71.4±3.9	77.6±2.1	71.8±3.5
*k*NN-FT	62.2±1.2	65.6±0.8	69.4*±2.3	70.8±1.9	77.2±1.5	70.6±3.4
*k*NN ranking	57.0	60.0	40.0	73.0*	77.0	49.0
Weighted *k*NN	57.0	60.0	37.0	73.0*	77.0	47.0
CE *k*NN$^+$ + SRC	59.8±1.8	**68.4***±0.8	65.6±1.6	68.8±3.9	76.8±0.7	67.6±2.8
PretRand	59.6±5.1	63.2±2.9	71.0*±0.6	72.2*±2.0	77.6±2.2	71.4±3.7
OT	65.4*±1.5	66.6±1.0	70.0*±2.8	71.4±5.2	73.6±3.6	**74.6***±2.9
OT^{NN}	**65.6***±2.9	67.4*±1.6	**71.6***±1.4	73.2*±0.7	73.8±2.3	72.6*±3.1
OT$^{NN}_{\mathtt{pre\text{-}select}}$	64.2±1.5	67.0±2.1	**71.6***±2.7	72.6*±1.0	75.4±1.4	73.2*±1.9
OT^{NN} + sloss	62.8±2.2	**68.4***±0.8	69.2*±3.2	**73.8***±1.6	76.8±1.9	73.4*±0.8
OT$^{NN}_{\mathtt{pre\text{-}select}}$ + sloss	65.2*±1.7	66.6±1.6	70.2*±3.7	72.2*±1.3	77.2±1.3	**74.6***±2.5

Table 2: F1 score (±std-dev) on the target corpus. The last four are the proposed OT^{NN} variants. **Bold** denotes the best, underline denotes the second-best scores in each column. * denotes the significantly improved scores compared to Seq-FT using the McNemar test (Dror et al., 2018; McNemar, 1947).

six respective settings of Table 2 are substantially higher than those from Target-FT. Furthermore, while the baseline methods show inconsistent performance across different settings, the proposed OT^{NN} variants yield the best performance in five out of six cases and the second-best in three cases. The baselines of Mixed-FT, *k*NN variants and CE *k*NN$^+$ achieve significant improvements compared to the vanilla Seq-FT for only 1 case, and PretRand achieves it for 2 cases. OT^{NN} variants, on the other hand, yield significant improvements in most cases; for instance, OT^{NN} has significantly improved scores in 5 out of 6 cases. Besides, the best scores from OT^{NN} variants improve over OT in 5 settings, while staying on par with OT in the remaining setting. This demonstrates that incorporating neighborhood information results in a more effective transfer.

When *Vidgen*$_{\text{src}}$ is used for transferring knowledge to *Ethos*$_{\text{tar}}$, Seq-FT yields the highest score (79.8). This is apparently because *Vidgen*$_{\text{src}}$ comprises a wide range of hateful forms directed towards different social groups. Since *Ethos*$_{\text{tar}}$ also involves hate against a variety of social groups, pretraining on all the source instances from *Vidgen*$_{\text{src}}$ for transfer learning, instead of training with the nearest neighbors, seems to be more helpful in this case. However, this is not the case when the transfer occurs from *Ethos*$_{\text{src}}$ to *Vidgen*$_{\text{tar}}$. This is likely because the *Vidgen* corpus involves adversarial instances that can easily fool an HS detection system trained on a different corpus. Besides, *Ethos*$_{\text{src}}$ has a subset of hateful forms and social groups covered by *Vidgen*. Therefore, a nearest neighborhood framework for transferring knowledge from *Ethos*$_{\text{src}}$ to *Vidgen*$_{\text{tar}}$ yields an improved performance, the highest score being 73.8 obtained by OT^{NN} + sloss, compared to 70.8 from Seq-FT.



(a) *Ethos*$_{\text{src}}$ to *Vidgen*$_{\text{tar}}$ (b) *Waseem*$_{\text{src}}$ to *Ethos*$_{\text{tar}}$

Figure 1: Performance with different sizes of the target train set. The total number of labeled instances available from the target is mentioned within the brackets, where the remaining instances are used as the target validation set.

Varying the size of X^t: We vary the size of the labeled target corpus available for training. We illustrate the cases of transferring knowledge from *Ethos*$_{\text{src}}$ to *Vidgen*$_{\text{tar}}$ in Figure 1(a), and from *Waseem*$_{\text{src}}$ to *Ethos*$_{\text{tar}}$ in Figure 1(b), with different OT^{NN} variants. For *Vidgen*$_{\text{tar}}$, we sample 300, 500, 700, and 900 instances. We use 80% for training, given by 240, 400, 560, and 720 instances, respectively, and the remaining 20% for validation. Since the *Ethos* corpus is small, we sample only 300, 500, and 700 instances as *Ethos*$_{\text{tar}}$, with the same proportions for training and validation. The target test set remains the same as in Table 1 for different training sizes. We observe that the OT^{NN} variants consistently improve the performance, with larger improvements obtained when the size of available target instances is lower. Mixed-FT, on the other hand, is inconsistent, and in some cases performs worse than Target-FT.

The improvements with OT^{NN} can be attributed to the fact that it can systematically *learn* the amount of transfer based on both the embedding distance and label consistency.

Target corpus	**Waseem$_{\mathrm{tar}}$**		**Vidgen$_{\mathrm{tar}}$**		**Ethos$_{\mathrm{tar}}$**	
Source corpus	**Vidgen$_{\mathrm{src}}$**	**Ethos$_{\mathrm{src}}$**	**Waseem$_{\mathrm{src}}$**	**Ethos$_{\mathrm{src}}$**	**Vidgen$_{\mathrm{src}}$**	**Waseem$_{\mathrm{src}}$**
OT^{NN} + sloss	$\underline{62.8}$±2.2	**68.4**±0.8	**69.2**±3.2	**73.8**±1.6	**76.8**±1.9	**73.4**±0.8
OT^{NN} + sloss (without ED)	**63.8**±1.3	65.8±1.7	$\underline{68.0}$±0.0	70.0±2.4	$\underline{76.4}$±0.8	$\underline{71.8}$±2.5
OT^{NN} + sloss (without LC)	62.0±2.1	$\underline{66.4}$±2.2	67.6±2.7	$\underline{72.4}$±1.4	75.2±2.6	67.8±3.9
OT$^{NN}_{\mathrm{pre\text{-}select}}$ + sloss	**65.2**±1.7	$\underline{66.6}$±1.6	**70.2**±3.7	**72.2**±1.3	**77.2**±1.3	**74.6**±2.5
OT$^{NN}_{\mathrm{pre\text{-}select}}$ + sloss (without ED)	$\underline{64.4}$±1.5	**67.6**±1.4	$\underline{67.6}$±4.3	70.8±2.3	$\underline{75.6}$±2.7	$\underline{74.2}$±5.6
OT$^{NN}_{\mathrm{pre\text{-}select}}$ + sloss (without LC)	62.2±2.6	63.8±1.5	67.2±5.0	$\underline{71.8}$±1.5	74.6±4.1	67.2±5.2

Table 3: Ablation study without the Embedding Distance (ED) /Label Consistency (LC) losses. F1 (±std-dev) on low-resource target corpus. **Bold** denotes the best, underline denotes the second-best score for each OT^{NN} variant.

Target instance: rt i am not sexist but girls who smoke are a big turn off even if your supermodel status its still a turn off.	
Instances	**GT**
Nearest neighbors from source using SBERT	
i do not actually care much for looks but the inflated sense of self importance that girls seem to have nowadays is rather obnoxious.	hate
totally agree. i could never find a woman or man attractive to go out with. .... i could nt be in a relationship with them as i am aromantic.	non-hate
every guy i know at work smoke at least 10 fags a day and i call them out. do not like it then do not do it.	non-hate
wow the truth is i did not get turned on until you mentioned cigar smoking men pissing on other men	non-hate
women have no redeeming features	hate
i am not sexist but women are dumb	hate
weed is stoner trash. if you are not a stoner guy, smoking this estrogen-filled garbage makes you a honorary one.	non-hate
maybe i am sexist because women are dumb	hate
women are less attracted to looks. they are far more attracted to high status and a dominant personality. the good news is you can acquire both of these.	non-hate
women without big t*ts are not attractive	hate
Nearest neighbors from source using OT^{NN}	
i am not sexist but women are dumb	hate
maybe i am sexist because women are dumb	hate
thrilled to have been born as the superior male gender	hate
men are superior than women in every single aspect	hate
women of color are not pleasant to look at	hate
leadership positions should only be occupied by men. that is the way to get things done.	hate
all women are created as equal b*tches	hate
all women should be suppressed	hate
this is the fact feminism is girlf*rts you cant unsmell	hate
women are devious	hate

Table 4: Qualitative analysis of an example with its top 10 nearest neighbors extracted using the SBERT and the learned OT^{NN} representations, where the source is *Vidgen*$_{\mathrm{src}}$ and the target is *Waseem*$_{\mathrm{tar}}$; GT: Ground Truth class.

5.2 Ablation Study

To analyze the importance of the constituent losses in OT^{NN}, we present an ablation study in Table 3 by removing the ED and the LC losses from the OT^{NN} variants. The best performances for each variant are obtained in 5 out of 6 cases when both the ED and the LC losses are incorporated. Besides, the second-best performances are obtained, in most cases, when we remove the ED loss. This suggests that while both losses are essential for an effective transfer, the LC loss contributes more towards the final performance than the ED loss.

5.3 Analysis of OT^{NN} Representations

We analyze the effect of training with OT^{NN} on the representation space by extracting the nearest neighbors of target instances. We rank these neighbors with cosine similarity over the learned OT^{NN} representations and check their ground truth classes. We compare them with the nearest neighbors obtained using SBERT representations. Table 4 contains an example of a hateful instance from *Waseem*$_{\mathrm{tar}}$, and its top 10 nearest neighbors from *Vidgen*$_{\mathrm{src}}$. We observe that the neighbors retrieved using the SBERT representations belong to both hate and non-hate classes. This is because SBERT is optimized mainly for semantic similarity, while they are sub-optimal in differentiating hateful instances from non-hateful ones. On the other hand, the neighbors obtained from OT^{NN} representations indicate that OT^{NN} brings instances across corpora, which are both semantically similar (the topic of women) and belong to the same class closer in the representation space, compared to those belonging to the opposite class.

In addition, we study the effect of the OT^{NN} representations by performing a simple majority voting of the top k nearest neighbors retrieved from the source with SBERT versus OT^{NN}. Figure 2 demonstrates the performance obtained on the target test set. Here the neighbors from the two representation spaces are ranked using cosine similarities. We can see that majority voting using the OT^{NN} representations achieves higher performance compared to that using the SBERT representations for different numbers of neighbors.

6 Conclusion and Future Work

In this work, we proposed a framework for transferring knowledge to a low-resource HS corpus by

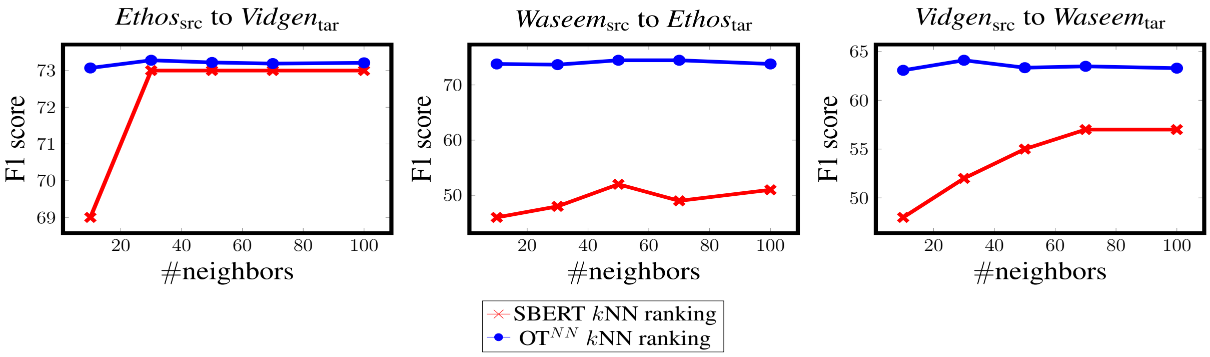


Figure 2: F1 using the majority voting of the k-Nearest Neighbors retrieved from SBERT and OT^{NN} representations.

incorporating neighborhood information with Optimal Transport. It allowed the model to flexibly learn the amount of transfer from the nearest neighbors based both on their proximity in a sentence embedding space and label consistency. Our framework yielded substantial improvements across HS corpora from varied platforms in low-resource settings. Besides, the qualitative analysis of its learned representations demonstrated that they incorporate both semantic and label similarities. This is different from sentence embedding representations, where semantically similar instances may have opposite labels.

Since our framework uses neighborhood information for transferring knowledge, it relies on the degree of proximity of the neighbors. However, if all of the source and target instances are very distant semantically, all the nearest neighbors from the source may have very low cosine similarity to the corresponding target instances. In such scenarios, the framework may yield limited improvements over the vanilla fine-tuning as the available neighborhood information would be much weaker. In such cases, the performance would mainly depend on the label consistency of the neighbors.

For future work, our framework can be explored for transferring knowledge from resource-rich languages, such as English, to low-resource languages. This can be done by extracting the cross-lingual neighbors using multilingual sentence embedding models like LaBSE (Feng et al., 2022). Besides, the framework can be applied for transferring knowledge in other text classification tasks, such as sentiment classification, bragging detection (Jin et al., 2022), etc., as the methodology is not restricted to only hate speech detection.

Ethical Considerations

The proposed approach intends to support more robust detection of online hate speech that can use the existing annotated resources for transferring knowledge to a resource with limited annotations. We acknowledge that annotating hateful content can have negative effects on the mental health of the annotators. The corpora used in this work are publicly available and cited appropriately in this paper. The authors of the respective corpora have provided detailed information about the sampling strategies, data collection process, annotation guidelines, and annotation procedure in peer-reviewed articles. Besides, the hateful terms and slurs presented in the work are only intended to give better insights into the models for research purposes.

Acknowledgements

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 are extremely grateful to Claire Gardent for taking time out to review the paper internally and Michel Olvera for his very helpful feedbacks regarding the work and for internally reviewing the paper. We would also like to thank the anonymous reviewers for their valuable feedbacks and suggestions.

References

Hind S. Alatawi, Areej Alhothali, and Kawthar Moria. 2021. Detection of hate speech using bert and hate speech word embedding with deep model. *ArXiv*, abs/2111.01515.

Ahmad Alwosheel, Sander van Cranenburgh, and Caspar G Chorus. 2018. Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of Choice Modelling*, 28:167–182.

Aymé Arango, Jorge Pérez, and Barbara Poblete. 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pages 45–54, New York, NY, USA. Association for Computing Machinery.

Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. 2022. Entropy-based attention regularization frees unintended bias mitigation from lists. In *Findings of the Association for Computational Linguistics: ACL2022 (Forthcoming)*. Association for Computational Linguistics.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 759–760, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Jean-David Benamou. 2003. Numerical resolution of an "unbalanced" mass transport problem. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 37(5):851–868.

Irina Bigoulaeva, Viktor Hangya, and Alexander Fraser. 2021. Cross-lingual transfer learning for hate speech detection. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 15–25, Kyiv. Association for Computational Linguistics.

Tulika Bose, Irina Illina, and Dominique Fohr. 2021. Generalisability of topic models in cross-corpora abusive language detection. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 51–56, Online. Association for Computational Linguistics.

Annisa Briliani, Budhi Irawan, and Casi Setianingsih. 2019. Hate speech detection in indonesian language on instagram comment section using k-nearest neighbor classification method. *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, pages 98–104.

Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. 2020. Graph optimal transport for cross-domain alignment. In *International Conference on Machine Learning*, pages 1542–1553. PMLR.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. 2017. Joint distribution optimal transportation for domain adaptation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3733–3742, Red Hook, NY, USA. Curran Associates Inc.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. 2018. DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 447–463.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Ashwin Geet D'Sa, Irina Illina, and D. Fohr. 2020. Bert and fastText embeddings for automatic detection of toxic speech. In *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, pages 1–5.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2021. Augmenting transformers with KNN-based composite memory for dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99.

Kilian Fatras, Thibault Séjourné, Rémi Flamary, and Nicolas Courty. 2021. Unbalanced minibatch optimal transport; Applications to domain adaptation. In *International Conference on Machine Learning*, pages 3186–3197. PMLR.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. 2021. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.

Komal Florio, Valerio Basile, Marco Polignano, Pierpaolo Basile, and Viviana Patti. 2020. Time of your hate: The challenge of time in hate speech detection on social media. *Applied Sciences*, 10(12).

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection. In *34th AAAI Conference on Artificial Intelligence*.

Goran Glavaš, Mladen Karan, and Ivan Vulić. 2020. XHate-999: Analyzing and detecting abusive language across domains and languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mali Jin, Daniel Preotiuc-Pietro, A. Seza Doğruöz, and Nikolaos Aletras. 2022. Automatic identification and classification of bragging in social media. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3945–3959, Dublin, Ireland. Association for Computational Linguistics.

Leonid V Kantorovich. 2006. On the translocation of masses. *Journal of Mathematical Sciences*, 133(4):1381–1382.

Mladen Karan and Jan Šnajder. 2018. Cross-domain detection of abusive language online. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 132–137, Brussels, Belgium. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. BERT-kNN: Adding a kNN search component to pretrained language models for better QA. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3424–3430, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Seonghyeon Lee, Dongha Lee, Seongbo Jang, and Hwanjo Yu. 2022. Toward interpretable semantic textual similarity via optimal transport-based contrastive sentence learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5969–5979, Dublin, Ireland. Association for Computational Linguistics.

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative studies of detecting abusive language on Twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 101–106, Brussels, Belgium. Association for Computational Linguistics.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:187 – 202.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Sara Meftah, Nasredine Semmar, Youssef Tamaazousti, Hassane Essafi, and Fatiha Sadat. 2021. Neural supervised domain adaptation by augmenting pre-trained models with random units. *ArXiv*, abs/2106.04935.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. ETHOS: A multi-label hate speech detection dataset. *Complex & Intelligent Systems*.

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2019. A BERT-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*, pages 928–940. Springer.

Michel Olvera, Emmanuel Vincent, and Gilles Gasso. 2021. Improving sound event detection with auxiliary foreground-background classification and domain adaptation. In *DCASE 2021-6th Workshop on Detection and Classification of Acoustic Scenes and Events*.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti. 2021. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Information Processing Management*, 58(4):102544.

Gabriel Peyré and Marco Cuturi. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Fabio Poletto, Valerio Basile, Cristina Bosco, Viviana Patti, and Marco Antonio Stranisci. 2019. Annotating hate speech: Three schemes at comparison. In *6th Italian Conference on Computational Linguistics, CLiC-it*.

Vincentius Riandaru Prasetyo and Anton Hendrik Samudra. 2022. Hate speech content detection system on twitter using k-nearest neighbor method. In *AIP Conference Proceedings*, volume 2470, page 050001. AIP Publishing LLC.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ankan Saha and Vikas Sindhwani. 2012. Learning evolving and emerging topics in social media: A dynamic NMF approach with temporal regularization. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, page 693–702, New York, NY, USA. Association for Computing Machinery.

Sheikh Muhammad Sarwar, Dimitrina Zlatkova, Momchil Hardalov, Yoan Dinkov, Isabelle Augenstein, and Preslav Nakov. 2022. A neighborhood framework for resource-lean content flagging. *Transactions of the Association for Computational Linguistics*, 10:484–502.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it blend? Blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Melbourne, Australia. Association for Computational Linguistics.

Steve Durairaj Swamy, Anupam Jamatia, and Björn Gambäck. 2019. Studying generalisability across abusive language detection datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 940–950, Hong Kong, China. Association for Computational Linguistics.

Kyle Swanson, Lili Yu, and Tao Lei. 2020. Rationalizing text matching: Learning sparse alignments via optimal transport. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5609–5626, Online. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, Online. Association for Computational Linguistics.

Cédric Villani. 2009. Optimal transport: Old and new. volume 338. Springer.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 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)*, pages 602–608, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7361–7373, Online. Association for Computational Linguistics.

Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7.

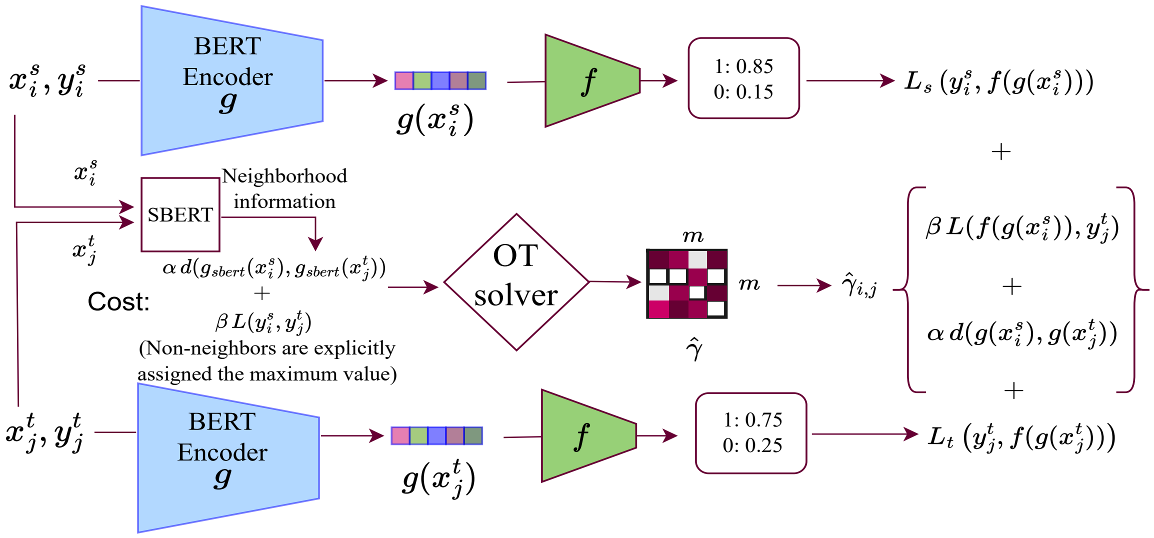
A Illustration of OT^{NN}

Figure 3 presents an illustration of the proposed OT^{NN} training strategy.

B Corpus Details

The corpora used in our experiments are collected during different time periods, with different sampling strategies across varied online platforms. Following are some additional details about the corpora discussed in Section 4.1.

Waseem: This Twitter corpus, provided by Waseem and Hovy (2016), is sampled mainly using keywords containing common terms and slurs associated with hate against sexual, gender, religious, and ethnic minorities. It originally has three



x_i^s, y_i^s
BERT Encoder g
$g(x_i^s)$
f
1: 0.85
0: 0.15
$L_s\,(y_i^s, f(g(x_i^s)))$
+
x_i^s
SBERT
Neighborhood information
x_j^t
Cost:
$\alpha\, d(g_{sbert}(x_i^s), g_{sbert}(x_j^t))$
+
$\beta\, L(y_i^s, y_j^t)$
(Non-neighbors are explicitly assigned the maximum value)
OT solver
m
m
$\hat{\gamma}$
$\hat{\gamma}_{i,j}$
$\beta\, L(f(g(x_i^s)), y_j^t)$
+
$\alpha\, d(g(x_i^s), g(x_j^t))$
+
x_j^t, y_j^t
BERT Encoder g
$g(x_j^t)$
f
1: 0.75
0: 0.25
$L_t\,(y_j^t, f(g(x_j^t)))$

Figure 3: Illustration of the training strategy in OT^{NN}. Even though the BERT encoder g and the classifier f are shared by both corpora, they are illustrated twice for better clarity by representing the two corpora separately. The presented softmax values obtained from f are simply examples provided for illustration. The figure is inspired by Damodaran et al. (2018).

classes: 'sexism', 'racism', and 'none'. However, since it is made available as tweet IDs, we observe that a major portion of the racist comments is removed by Twitter, and we could retrieve only 20 instances labeled as 'racism'. Due to the difference in the labeling schemes across corpora, we follow other prior work for cross-corpus evaluations (Wiegand et al., 2019; Swamy et al., 2019), and merge the original 'racism' and 'sexism' classes into the class *hate*. In total, we obtain 10.9K tweets, of which 26.8% instances belong to the *hate* class. A major portion of hateful comments in this corpus is subtle and do not involve profane words (Wiegand et al., 2019).

Vidgen: The *Vidgen* corpus (Vidgen et al., 2021) is dynamically generated using a human-and-model-in-the-loop process that is aimed at making the corpus robust to different forms of hate. These hateful forms are derogation, animosity, threatening language, support for hateful entities, and dehumanization. It covers hate directed against diverse social groups like blacks, women, jews, muslims, immigrants, arabs, disabled people, trans people, and others. It has adversarial content that can mislead the HS detection systems trained on other corpora, such as '*all women are created equal, except chinks*', '*the tea cup is bigger than the f*cking teapot*'. We use its version v0.2.3 containing a total of 41144 instances, where duplicates are removed, of which 53.9% is labeled as *hate*.

Ethos: This corpus (Mollas et al., 2022) consists of 998 comments from YouTube and Reddit, sampled using an active learning strategy, which ensures both diversity and balance with respect to different hateful aspects defined. One of the ways they ensure this is by keeping only one instance of multiple comments with similar meanings. For example, the comments '*I hate white people*' and '*I hate whites*' (Mollas et al., 2022) are similar, and only one of them is added. It includes hate directed towards diverse identities, such as gender, race, national origin, disability, religion, and sexual orientation. In this work, we use the binary version of this corpus with 43.4% *hate* instances.

C Data Preprocessing

We pre-process the corpora by removing the URLs, splitting the hashtags into constituent words using CrazyTokenizer[1], expanding contractions (e.g. i'll to i will), and removing the rarely occurring Twitter handles and numbers. We finally convert the instances into lower case.

D Implementation Details

For implementing the proposed OT^{NN} framework, we fine-tune the pre-trained BERT-base uncased model, implemented by Hugging Face (Wolf et al., 2020), having 110 million parameters, with the

[1] https://redditscore.readthedocs.io

Target corpus	**Waseem$_{\mathtt{tar}}$**		**Vidgen$_{\mathtt{tar}}$**		**Ethos$_{\mathtt{tar}}$**	
Source corpus	**Vidgen$_{\mathtt{src}}$**	**Ethos$_{\mathtt{src}}$**	**Waseem$_{\mathtt{src}}$**	**Ethos$_{\mathtt{src}}$**	**Vidgen$_{\mathtt{src}}$**	**Waseem$_{\mathtt{src}}$**
Seq-FT	63.2±2.1	65.0±1.1	67.0±2.2	70.8±3.9	79.8±0.7	70.2±3.1
k = 10	59.8±1.8	68.4±0.8	65.6±1.6	68.8±3.9	76.8±0.7	67.6±2.8
k = 20	61.2±1.5	67.6±1.5	64.8±1.6	69.2±3.2	76.8±1.0	67.4±3.3
k = 30	60.3±1.6	68.1±1.0	64.4±1.9	69.9±2.8	76.8±0.5	68.5±1.7
k = 40	61.6±1.6	68.6±1.4	64.6±1.0	70.8±3.5	76.2±1.2	68.2±2.6
k = 50	60.8±2.0	68.8±0.7	62.8±2.6	68.4±4.8	75.8±0.4	68.4±0.5

Table 5: Performance of CE kNN$^+$ + SRC with different neighborhood sizes, compared with Seq-FT. F1 score (±std-dev) is reported on the low-resource target corpus with 400 labeled training instances (total 500 labeled instances from the target) available.

joint distribution OT framework[2]. We encode an instance into the embedding space by obtaining the representations of the [CLS] token from the last hidden layer of BERT, which is a 768-dimensional vector in the BERT-base. We fine-tune the BERT model end-to-end for the classification task. Therefore, the [CLS] representations are the fine-tuned BERT representations. For incorporating the neighborhood information, we use the pre-trained SBERT sentence embeddings from 'all-mpnet-base-v2'[3] model, which is a sentence transformer model. For computing γ, we use the entropic regularized unbalanced OT solver using the Python Optimal Transport package[4] (Flamary et al., 2021) at the mini-batch level.

For the baselines of kNN-FT, kNN ranking, weighted kNN and the OT^{NN} variants, we select the number of neighbors (k) from the range {10, 30, 50, 70, 100, 200, 300, 400, 500} through tuning over the corresponding target validation sets with respect to the F1 score of the hate class with a random seed. We set α = 0.05 and β = 10 in Equation 3 and 5, and θ_s = 1 for OT^{NN} / OT$^{NN}_{\text{pre-select}}$ + sloss and θ_t = 10 in Equation 4, 6 and 7 for all the experiments. For OT^{NN} without sloss, we set θ_s = 0.

For CE kNN$^+$ + SRC, we perform experiments with the implementation provided to us by the authors and report the results for the neighborhood size of 10 in Table 2. Even though Sarwar et al. (2022) use 10 as the neighborhood size in their task of transfer learning in a cross-lingual set-up, we experiment with different neighborhood sizes (k values). The results are reported in Table 5. However, we could not increase the neighborhood size beyond 50 because of resource constraints. This is because a mini-batch in their framework comprises a query instance from the target and all its k neighbors from the source. Thus, the number of neighbors is limited by the mini-batch size, which usually needs to be kept small when fine-tuning large language models like BERT. We can observe from Table 5 that the performances obtained with different neighborhood sizes are similar.

We implement PretRand ourselves following the description provided by Meftah et al. (2021). This approach is evaluated by the authors on the tasks of part-of-speech tagging, chunking, named entity recognition, and morphosyntactic tagging. Therefore, the approach uses a sequence labeling model with pre-trained word embeddings and a BiLSTM-based feature extractor. However, for a fair comparison with our approach, we use the pre-trained BERT model as the feature extractor instead of the BiLSTM model for the pre-trained units. For the randomly initialized units, we follow the approach and add a BiLSTM layer over the last hidden layer of the BERT model. We first fine-tune the pre-trained BERT model, without the randomly initialized units, on the source corpus. We then fine-tune the model with the additional randomly initialized units on the target corpus. We use the Adam optimizer with a learning rate of 5×10^{-5} for the pre-trained BERT parameters. For the randomly initialized units, we use the Adam optimizer with a learning rate of 1.5×10^{-2} following Meftah et al. (2021).

E Computational Efficiency

We present the per epoch training time of Mixed-FT and OT^{NN} variants for different settings of the source and target corpora in Table 6. Mixed-FT is a baseline that involves training the pre-trained BERT model on the combination of the source and target corpora. For every mini-batch of size m, there are m instances sampled from each of the source and target corpora (Equation 7). This is the same mini-batch sampling that is followed in

[2] https://github.com/bbdamodaran/deepJDOT

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[4] https://pythonot.github.io/gen_modules/ot.unbalanced.html#ot.unbalanced.sinkhorn_unbalanced

Target corpus	**Waseem**$_{\texttt{tar}}$		**Vidgen**$_{\texttt{tar}}$		**Ethos**$_{\texttt{tar}}$	
Source corpus	**Vidgen**$_{\texttt{src}}$	**Ethos**$_{\texttt{src}}$	**Waseem**$_{\texttt{src}}$	**Ethos**$_{\texttt{src}}$	**Vidgen**$_{\texttt{src}}$	**Waseem**$_{\texttt{src}}$
Mixed-FT	17.8 m	0.4 m	4.7 m	0.5 m	14.0 m	4.7 m
OT^{NN}	18.9 m	0.4 m	5.1 m	0.6 m	14.2 m	5.0 m
OT$^{NN}_{\texttt{pre-select}}$	3.7 m	0.3 m	1.1 m	0.6 m	6.5 m	3.4 m
OT^{NN} + sloss	18.9 m	0.4 m	5.0 m	0.6 m	14.5 m	4.9 m
OT$^{NN}_{\texttt{pre-select}}$ + sloss	11.7 m	0.4 m	3.8 m	0.6 m	5.5 m	3.9 m

Table 6: Per epoch training time in minutes for different settings.

OT^{NN}. We use one Nvidia GTX 1080 Ti GPU for our experiments. We can observe that OT^{NN} results in approximately the same computation time as taken by Mixed-FT in most of the settings as it does not change the model architecture, but only introduces a new training strategy. With the 'pre-select' variant, the computation time gets further reduced in a few settings. This is because, in this variant, the model only gets trained on a subset of pre-selected source instances based on the neighborhood size.