Reinforcement Guided Multi-Task Learning Framework for Low-Resource Stereotype Detection

Rajkumar Pujari[1] and **Erik Oveson**[2] and **Priyanka Kulkari**[2] and **Elnaz Nouri**[3]
[1]Purdue University * [2]Microsoft, Redmond [3]Microsoft Research, Redmond
rpujari@purdue.edu {erikov,priyak,elnouri}@microsoft.com

Abstract

As large Pre-trained Language Models (PLMs) trained on large amounts of data in an unsupervised manner become more ubiquitous, identifying various types of bias in the text has come into sharp focus. Existing '*Stereotype Detection*' datasets mainly adopt a diagnostic approach toward large PLMs. Blodgett et al. (2021a) show that there are significant reliability issues with the existing benchmark datasets. Annotating a reliable dataset requires a precise understanding of the subtle nuances of how stereotypes manifest in text. In this paper, we annotate a focused evaluation set for '*Stereotype Detection*' that addresses those pitfalls by de-constructing various ways in which stereotypes manifest in text. Further, we present a multi-task model that leverages the abundance of data-rich neighboring tasks such as hate speech detection, offensive language detection, misogyny detection, etc., to improve the empirical performance on '*Stereotype Detection*'. We then propose a reinforcement-learning agent that guides the multi-task learning model by learning to identify the training examples from the neighboring tasks that help the target task the most. We show that the proposed models achieve significant empirical gains over existing baselines on all the tasks.

1 Introduction

Following Peters et al. (2018) and then Vaswani et al. (2017), several empirically effective large pre-trained language models (PLMs) (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019; Lewis et al., 2020; Brown et al., 2020) have been proposed. Their empirical success on several existing tasks has resulted in them being ubiquitously used in many NLP applications that interact with humans on a daily basis (Olteanu et al., 2020). As they are trained in an unsupervised manner on large amounts of arbitrary web data, harmful language and bias creeps into their output. This, in turn, is reflected in the applications that are built on top of them and further propagated in society through the use of those applications (Berk, 2017). This issue motivates the need for a two-pronged solution: 1) to diagnose and de-noise the bias in the PLMs and 2) to identify and regulate harmful text externally at the output level. Progress on the first front is necessary to de-bias the PLMs and prevent subtle biases that creep in through their distributed representations which can often only be recognized in retrospect. This results in deep distrust of such systems among general population. The second front is paramount to detect and potentially block harmful output and also ensure that future iterations and novel proposals of the language models don't take spurious paths to gaming the de-biasing techniques. In this work, we focus on identifying harmful text with stereotypical associations externally.

S_1: She may or may not be a jew but she's certainly cheap! (*insult, stereotype*)
S_2: Burn in hell, you Asian bastard! (*abuse, stereotype*)
S_3: Asians are good at math. (*stereotype*)
S_4: My African-American friend owns a watermelon patch. (*stereotype*)

There exist several types of harmful language such as hate-speech, misogyny, stereotypes, abuse, threats, insult etc,. Each type of offensive language has subtle linguistic nuances that are specific to the type of offensive language. Often, offensive text contains multiple types of offense. From the examples above, consider S_1 and S_2. Both, consist of multiple modes of offense. While S_3 is purely a stereotype, it is still undesirable to be perpetuated.

Cardwell (1996) defines *stereotype* as a "*fixed, over-generalized belief about a particular group or class of people*". Stereotypes differ from other types of offensive text in two key aspects: (1) they require knowledge of their existence in the society to be identified, and (2) they might also often

*This work is a part of summer internship at Microsoft Research, Redmond

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6703 - 6712

express positive sentiment about the target group. Although some stereotypes ostensibly express positive sentiment towards the target group, they are still undesirable as they propagate false biases in the society and are offensive to the target group. Consider sentences S_3 and S_4 from above examples. While S_3 expresses positive sentiment, it is still false and undesirable. S_4 requires knowledge of that particular stereotype's history to understand its offensive nature. Requiring prior knowledge makes annotating data for the task of '*Stereotype Detection*' harder, as annotators are unlikely to be aware of all the stereotypes that exist in the society. (Czopp, 2008).

Two recent works have proposed pioneering diagnostic datasets for measuring stereotypical bias of large PLMs (Nadeem et al., 2020; Nangia et al., 2020). But, Blodgett et al. (2021b) has demonstrated that these datasets suffer from two major types of issues: (1) conceptual: include harmless stereotypes, artificial anti-stereotypes, confusing nationality with ethnicity etc, and (2) operational: invalid perturbations, unnatural text, incommensurable target groups etc,. In addition, diagnostic datasets also suffer from lack of sufficient coverage of subtle nuances of manifestations of stereotypes in text. This makes them less suitable for training an effective discriminative classifier. Hence, we undertake a focused annotation effort to create a fine-grained evaluation dataset. We mainly aim to alleviate the *conceptual* issues of anti- vs. non-stereotypes, containing irrelevant stereotypes and *operational* issues of unnatural text, invalid perturbations. We achieve this by a mix of (1) selecting more appropriate data candidates and (2) devising a focused questionnaire for the annotation task that breaks down different dimensions of the linguistic challenge of '*Stereotype Identification*'. Collecting real-world data from the social forum Reddit for annotation also results in better coverage of subtle manifestations of stereotypes in text.

Although *stereotypes* differ from other types of *offensive language* in multiple ways, they also overlap to a significant extent. Often, various types of offensive text such as abuse, misogyny and hate speech integrally consists stereotypical associations. Abundance of high-quality annotated datasets are available for these neighboring tasks. We leverage this unique nature of *Stereotype Detection* task to propose a multi-task learning framework for all related tasks. As the overlap between the tasks is only partial, we then propose a reinforcement learning agent that learns to guide the multi-task learning model by selecting meaningful data examples from the neighboring task datasets that help in improving the target task. We show that these two modifications improve the empirical performance on all the tasks significantly. Then, we look more closely at the reinforcement-learning agent's learning process via a suite of ablation studies that throw light on its intricate inner workings. To summarize, our main contributions are:

1. We devise a focused annotation effort for *Stereotype Detection* to construct a fine-grained evaluation set for the task.
2. We leverage the unique existence of several correlated neighboring tasks to propose a reinforcement-learning guided multitask framework that learns to identify data examples that are beneficial for the target task.
3. We perform exhaustive empirical evaluation and ablation studies to demonstrate the effectiveness of the framework and showcase intricate details of its learning process.[1]

2 Related Work

With the rise of social media and hate speech forums online (Phadke and Mitra, 2020; Szendro, 2021) offensive language detection has become more important that ever before. Several recent works focus on characterizing various types of offensive language detection (Fortuna and Nunes, 2018; Shushkevich and Cardiff, 2019; Mishra et al., 2019; Parekh and Patel, 2017). But, works that focus solely on *Stereotype Detection* in English language are scarce. This is partly because stereotypes tend to be subtler offenses in comparison to other types are offensive languages and hence receive less immediate focus, and in part due to the challenge of requiring the knowledge of the stereotype's existence in society to reliably annotate data for the task. We approach this problem by breaking down various aspects of stereotypical text and crowd-sourcing annotations only for aspects that require linguistic understanding rather than world-knowledge.

Few recent works have focused solely on *stereotypes*, some proposing pioneering diagnostic datasets (Nadeem et al., 2020; Nangia et al.,

[1]Our code and data is available at https://github.com/pujari-rajkumar/rl-guided-multitask-learning

Examples
1. Ethiopians like stew (*Explicit Stereotype*)
2. The lawyer misrepresented the situation and tricked the person (*Implicit Stereotypical Association*)
3. Jews spend money frivolously (*Anti-Stereotypes*)
4. There is an Asian family that lives down the street (*Non-Stereotypes*)

Table 1: Examples of Various Categories of Text with Stereotypical Associations

2020) while others worked on knowledge-based and semi-supervised learning based models (Fraser et al., 2021; Badjatiya et al., 2019) for identifying stereotypical text. Computational model based works either use datasets meant for other tasks such as hate speech detection etc, or focus mainly on the available diagnostic datasets modified for classification task. But, diagnostic datasets suffer from lack of sufficient coverage of naturally occurring text due to their crowd-sourced construction procedure (Blodgett et al., 2021b). We address these issues in our work by collecting natural text data from social forum Reddit, by mining specific subreddits that contain mainly subtle stereotypical text.

Multi-task learning (Caruana, 1997), can be broadly classified into two paradigms (Ruder, 2017): hard parameter sharing (Caruana, 1997) and soft parameter sharing (Yang and Hospedales, 2016; Duong et al., 2015). We implement hard-parameter sharing based multi-task model for our experiments.

Given the low-resource setting on *Stereotype Detection* task, semi-supervised data annotation is one plausible solution for the problem. Several recent works have also been focusing on reinforcement-learning guided semi-supervision (Ye et al., 2020; Konyushkova et al., 2020; Laskin et al., 2020). Ye et al. (2020), in particular, work with a single-task and unsupervised data to generate automated-annotations for new examples. In contrast, we use the data from neighboring tasks with different labels for multi-task learning and apply an RL agent to select examples for training the neighboring task in such a way that benefits the target task the most.

3 Our Dataset

As Blodgett et al. (2021a) demonstrates, existing diagnostic datasets such as *Stereoset* (Nadeem et al., 2020) and CrowS-Pairs (Nangia et al., 2020) suffer from several conceptual and operational issues. Additionally diagnostic datasets, by nature, also suffer from lack of coverage of subtle manifestations of stereotypes in text. But, this is mandatory in the dataset used to train an effective discriminatory classifier that can generalize well to identifying stereotypical associations in open domain use cases. This is often the case in the examples encountered in downstream applications that interact with humans on a daily basis.

Nadeem et al. (2020) and Nangia et al. (2020) both follow the following dataset construction procedure: they provide a prompt word or phrase, which is the target group of the stereotype, and ask the crowd workers to construct a stereotypical sentence which includes the provided prompt. Then, they either flip the identity of the target group (Nangia et al., 2020), the associated characteristic/action of the group or both (Nadeem et al., 2020) to obtain negative (*anti-stereotypes*) examples.

Although this process results in good coverage of existing stereotypes within society due to the variety of prompts provided, it also results in data examples that either explicitly state the stereotype or are anti-stereotypes that are mostly contrived sentences and are unlikely to occur in regular discourse. An example from *Stereoset* dataset (Nadeem et al., 2020) is shown below to illustrate this point. 1 is an explicitly states stereotype where as 2 and 3 are unlikely to occur in regular discourse.

1. He is an Arab. Must be a terrorist. (*stereotype*)
2. He is an Arab. Must be a pacifist. (*anti-stereotype*)
3. He is an Arab. Dog needs walk. (*unrelated*)

We address this issue by additionally annotating text from actual discussions from Reddit to create a more representative dataset for the task. We use the collected examples for a crowd-sourced annotation task to build the dataset. The annotation task is explained in further detail below. We use the prompts from Nadeem et al. (2020) and Nangia et al. (2020) to collect candidates from selected subreddits which are likely to contain a good coverage of stereotypical and non-stereotypical text. We also use the explicit stereotypes from the *Stereoset* and *CrowS Pairs* datasets as candidates. Then, we design a focused annotation task for these candidates

to identify various types of Stereotypes.

3.1 Construction Procedure

We make the key observation that there are three broad classes of text that contain potentially harmful over-simplified beliefs: 1) text that is mainly intended to express a stereotype (*explicit stereotype*), 2) text that is not mainly intended to convey a stereotype but nevertheless propagates a stereotypical association (*implicit stereotypical association*), and 3) text that expresses an over-simplified belief which is *not* that widely-known, hence is not considered stereotypical (*anti-stereotype*). In addition, there is 4) text that doesn't contain any over-simplified beliefs about a group of people. We call this type of text as *non-stereotypes*. Examples of different types text are shown in table 1.

Nadeem et al. (2020) and Nangia et al. (2020) are mainly made up of *explicit-stereotypes* and *anti-stereotypes*. They lack coverage of *implicity stereotypical associations* and *non-stereotypes* due to their construction process and the nature of diagnostic datasets. These are necessary to build a task-representative classification dataset. Hence, in our annotation task we aim to add non-stereotypical examples that contain the same prompts as the ones that are used to create the stereotypical examples. To this end, we collect candidate examples from two subreddit forums */r/Jokes* and */r/AskHistorians*. We observe that */r/Jokes* consists of a high percentage of text with stereotypical associations (both *explicit* and *implicit* stereotypes) and */r/AskHistorians* tends to consist mainly factual text that is carefully constructed to avoid stereotypical associations. We collect examples that contain the prompts provided in the Nadeem et al. (2020) dataset as candidates for annotation. We also use the explicit stereotypes from *Stereoset* and *CrowS-Pairs* datasets for annotation. We perform annotation using *Amazon Mechanical Truk*. For each candidate sentence, we ask the annotators to answer the following questions:

1. Is there an over-simplified belief about a particular type of person "intentionally" expressed in the text?
2. Is there an "unintentional", widely-known stereotypical association present in the text?
3. Does the sentence seem made up (unlikely to occur in regular discourse)?

Each example is annotated by three annotators and we use the majority answer as the gold label. This annotation allows us to separate the text into one of the above 4 categories. Our dataset consists of 742 explicit stereotypes, 282 implicit stereotypes and $1,197$ non-stereotypes. We show the summary statistics of the annotated dataset in table 2.

3.2 Ethics Statement

We conducted a qualification test to select workers based on their performance. The workers were paid a bonus of USD 0.10 for taking the qualification text. We paid USD 0.25 for a batch of 10 examples, each batch taking 45-60 seconds on average. This amounts to USD $15 - 20$/hour. We displayed a warning on the task that said that the task might contain potentially offensive language. We didn't collect any personal identifying information of the workers other than their worker ID for assigning qualifications. We restricted the workers location to the USA with minimum of $5,000$ approved HITs and 98% HIT approval rate.

Data Type	Size
Explicit Stereotypes	742
Implicit Stereotypes	282
Non-Stereotypes	1,197
Total Examples	2,221

Table 2: Summary Statistics of Annotated Dataset

4 Model

As discussed in section 1, high-quality gold data for *Stereotype Detection* is scarce. But, several tasks with correlating objectives have abundance of high-quality annotated datasets. We observe that several tasks under the general umbrella of *Offensive Language Detection* such as *Abuse Detection*, *Hate Speech Detection* & *Misogyny Detection* often include text with stereotypical associations, as demonstrated in examples S_1 and S_2 in section 1. We call these tasks *neighboring tasks*. We leverage the neighboring task datasets to improve the performance on the low-resource setting of *Stereotype Detection*. First, we propose a multi-task learning model for all the tasks. Then, we make the key observation that *"all examples from the neighboring tasks are* ***not*** *equally useful for the target task"* as the objectives only overlap partially. Further, we propose a reinforcement-learning agent, inspired from Ye et al. (2020), that learns to select data examples from the neighboring task datasets which are most relevant to the target task's learning ob-

jective. We guide the agent via reward assignment based on shared model's performance on the evaluation data of the target task. We experiment both the settings with 4 popular large PLMs as base classifiers and demonstrate empirical gains using this framework.

In subsection 4.1, we describe the multi-task learning (MTL) model followed by the Reinforcement Learning guided multi-task learning model (RL-MTL) in subsection 4.2. Then, in subsection 5.1, we describe the baseline classifiers we use for our experiments.

4.1 Multi-Task Learning Model

The motivation behind our Multi-Task Learning model is to leverage the transfer learning gains from the neighboring tasks to improve the target task. As the tasks have partially overlapping objectives, solving the selected neighboring tasks effectively requires an understanding of largely similar linguistic characteristics as the target task. Hence, leveraging the intermediate representations of the text from the neighboring task to boost the classifier is expected to benefit the target task.

Following this motivation, our proposed multi-task model consists of a fixed PLM-based representation layer, followed by shared parameters that are common for all the tasks. Then, we add separate classification heads for each task. We implement hard parameter sharing (Caruana, 1997; Ruder, 2017) in our model. The shared parameters compute intermediate representations for the text input. These intermediate representations are shared by all the tasks. Parameters for the shared representation layers are first optimized by training on the neighboring tasks. Then, they are leveraged as a more beneficial parameter initialization for training on the target task data.

The input to the multi-task model is the text of the data example and a task ID. Output of the model is predicted label on the specified task. Each task in the model could either be a single-class classification task or a multi-label classification task. Classification heads for single-class classification tasks have a softmax layer after the final layer. Multi-label tasks have a sigmoid layer for each output neuron in the final layer of the classification heads.

First, we jointly train the model on each of the neighboring tasks in a sequential manner. Then, we train the multi-task model on the target task and evaluate it on the test set of the target task.

4.2 Reinforcement Learning Guided MTL

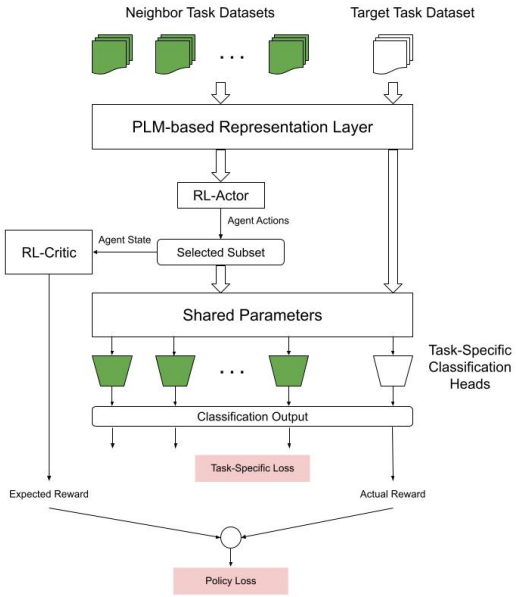

Figure 1: Reinforcement-guided Multi-task Learning Model for Low-Resource Classification Tasks with Correlated Neighboring Tasks

The RL-guided multi-task model has an additional RL agent on top of the MTL model to select examples from the neighboring task datasets that would be used to train the shared classifier. Key intuition behind the introduction of the RL agent is that, *not all data examples from the neighbor task are equally useful in learning the target task*. Architecture of the RL-guided MTL model is shown in figure 1.

Following the above observation, we employ the agent to identify examples that are useful for the target objective and drop examples that distract the classifier from the target task. The agent is trained using an actor-critic reinforcement paradigm (Konda and Tsitsiklis, 2000). For each example in the neighbor task, the *Actor* decides whether or not to use it for training the shared classifier. *Critic* computes the expected reward based on *Actor*'s actions for a mini-batch. Upon training using the selected examples, we then assign reward to the agent by evaluating the performance of the shared classifier on the target task. If the F_1 scores on the valuation set for b mini-batches, each of size z, are $\{F_1^0, F_1^1, \ldots, F_1^b\}$ and expected rewards predicted by the critic are $\{e_0, e_1, \ldots, e_b\}$, then the policy loss is computed as follows:

$$\hat{F}_1^i = \frac{F_1^i - \mu_{F_1}}{\sigma_{F_1} + \epsilon} \quad (1)$$

$$\text{p} = -\frac{1}{b}\Sigma_{i=1}^{b}(\hat{F}_1^i - e_i) \times \frac{1}{z}\Sigma_{j=1}^{z} log(P[a_j^i]) \quad (2)$$

$$\text{v} = \frac{1}{b}\Sigma_{i=1}^{b}\mathbb{L}_1\text{-loss}(1, \hat{F}_1^i) \quad (3)$$

$$\text{total loss} = \text{policy loss (p)} + \text{value loss (v)} \quad (4)$$

where ϵ is a smoothing constant, a_j^i is the action decided by the Actor for the j^{th} example of mini-batch i, μ_{F_1} and σ_{F_1} are mean and standard deviations of the macro-F_1 scores, respectively.

The algorithm for RL-guided Multitask learning is shown in algorithm 1. Input to the RL-MTL model is a set of neighboring task datasets and a target task dataset. Output is trained classifier $\mathbb{C}$. We initialize the parameters of the RL-MTL base classifier with the trained parameters of the MTL model. Later, we evaluate the impact of this initialization via an ablation study in section 7.1.

Algorithm 1 RL-Guided MTL

Require: Neighbor Datasets $\{\mathbb{N}_0, \mathbb{N}_1, \ldots, \mathbb{N}_d\}$, Target Dataset $\mathbb{T}$

Parameters: Policy Network $\mathbb{P}$ that includes Actor Network $\mathbb{A}$ and Critic Network $\mathbb{R}$

Select baseline classifier $\mathbb{C}$
for episode i = 1, 2, . . . , e **do**
  for neighbor dataset j = 1, 2, . . . , d **do**
    for mini-batch k = 1, 2, . . . , b **do**
      Actor Network $\mathbb{A}$ makes binary SELECT / REJECT decision for each example in $\mathbb{N}_{jk}$
      Critic Network $\mathbb{R}$ computes expected reward based on examples selected by Actor $\mathbb{A}$ = $E[r]^{ijk}$
      Train $\mathbb{C}$ on the SELECTED mini-batch subset $\mathbb{N}_{jk}^{SEL}$
      Evaluate on Target Dataset $\mathbb{T}$ and obtain F_1 on target dataset evaluation set F_1^{ijk}
    end for
    Use F_1^{ijk}s and $E[r]^{ijk}$s to compute loss according to equation 4
    Update parameters of $\mathbb{A}$ and $\mathbb{R}$
  end for
end for
return Trained classifier $\mathbb{C}$

5 Experiments

We perform experiments on *six* datasets in *three* phases. In the first phase, we experiment with PLM-based fine-tuned classifiers for each task as baselines. In the second phase, we experiment with all the tasks using the multi-task learning model described in section 4.1, with each PLM as a base classifier. In the third phase, we train the reinforcement-learning guided multi-task learning framework (section 4.2) for all the tasks with each of the PLMs as base classifier.

5.1 Base Classifiers

We select four popular PLMs as base classifiers for our empirical experiments, namely, BERT-base, BERT-large (Devlin et al., 2019), BART-large (Lewis et al., 2020) and XLNet-large (Yang et al., 2019). We use the implementations from Wolf et al. (2020)'s huggingface transformers library[2] for experimentation. We fine-tune a classification layer on top of representations from each of the PLMs as baseline to evaluate our framework.

5.2 Datasets

We use *six* datasets for our empirical evaluation, namely, Jigsaw Toxicity Dataset, Hate Speech Detection (de Gibert et al., 2018), Misogyny Detection (Fersini et al., 2018), Offensive Language Detection (Davidson et al., 2017), coarse-grained Stereotype Detection (combination of *Stereoset*, *CrowS-Pairs* and Reddit Data) and finally fine-grained Stereotype Detection Data (as described in section 3). We describe each dataset briefly below.

Hate Speech Detection (de Gibert et al., 2018) dataset consists of $10,944$ data examples of text extracted from Stromfront, a white-supremacist forum. Each piece of text is labeled as either *hate speech* or *not*.

Misogyny Detection (Fersini et al., 2018) dataset consists of $3,251$ data examples of text labeled with the binary label of being *misogynous* or *not*.

Offensive Language Detection (Davidson et al., 2017) dataset was built using crowd-sourced hate lexicon to collect tweets, followed by manual annotation of each example as one of *hate-speech*, *only offensive language* or *neither*. This dataset contains $24,783$ examples.

Coarse-Grained Stereotype Detection: We create this dataset by combining stereotypical examples from *Stereoset* and *CrowS-Pairs* datasets to get positive examples, followed by adding negative examples from the subreddit */r/AskHistorians*. We

[2] https://github.com/huggingface/transformers

do not use crowd sourced labels in this dataset. We use the labels from the original datasets. The dataset consists of $23,900$ data examples.

Fine-Grained Stereotype Detection: This dataset is the result of our annotation efforts in section 3. It consists of $2,221$ examples, each annotated with one of three possible labels: *explicit stereotype, implicit stereotype and non-stereotype*.

Jigsaw Toxicity Dataset[3] consists of $159,571$ training examples and $153,164$ test examples labeled with one or more of the *seven* labels: *toxic, severely toxic, obscene, threat, insult, identity hate, none*. We use this data only for training. We don't evaluate performance on this dataset.

6 Results

We present the results of the empirical evaluation tasks in table 3. In *Hate Speech Detection* task, we observe that RL-MTL learning results in significant improvements over all the baseline classifiers. Plain MTL model also improves upon the baseline classifiers except in the case on BART-large. The best model for this task is BERT-base + RL-MTL which achieves a macro-F1 score of 72.06 compared to 68.91 obtained by the best baseline classifier. Best MTL model obtains 69.78 F1.

For *Hate Speech and Offensive Language Detection* task, the respective numbers for baseline, MTL and RL-MTL models are 66.13, 68.57 and 68.97. The models achieve 74.16, 74.40 and 75.21 on *Misogyny Detection* task, respectively. In *Coarse-Grained Stereotype Detection* task, they achieve 65.71, 68.29 & 74.18, which is a significant gradation over each previous class of models. On our focus evaluation set of *Fine-Grained Stereotype Detection*, we achieve 61.36, 65.00 & 67.94 in each class of models. The results on this dataset are obtained in a zero-shot setting as we only use this dataset for evaluation.

7 Analysis & Discussions

In the first ablation study described in subsection 7.1, we study the importance of initializing RL-MTL model with the trained parameters of MTL model. Following that, we look into more detail about the usefulness of neighbor tasks on the target task via an ablation study.We describe these experiments in further detail in subsection 7.2.

[3] https://tinyurl.com/2vjmprnh

7.1 Impact of MTL Prior on RL-MTL

In our original experiments, we initialize the parameters of RL-MTL model with trained parameters from the MTL model. This allows the RL agent to begin from a well-optimized point in the parameter sample space. In this ablation study, we initialize the RL-MTL model from scratch to see how it impacts the performance of the RL-MTL model. We perform this experiment with BERT-base as base classifier. The performance of the RL- MTL model without initialization drops to 70.23 on HS task, 67.23 on HSO task, 71.10 on MG task, 60.42 on CG-ST task and 57.32 on FG-ST task. The respective numbers for the MTL initialized model are 72.06, 68.97, 74.78, 74.18 and 65.72. Initialization has biggest impact on the *Coarse-* and *Fine-Grained Stereotype Detection* tasks. Overall, initialization with MTL trained parameters results in a better convergence point for the RL-MTL model.

7.2 Neighbor-Task Ablation Study

In this task, we aim to study the neighbor tasks that are most useful for each target task. For each dataset, we train RL-MTL framework with only one other neighbor dataset. We see which task yields biggest improvement for each target task. We experiment with various combinations of datasets for this dataset. Results for this ablation study are shown in table 4. All experiments in this ablation study are performed using BERT-base as the base classifier.

Results in table 4 show that for both *Hate Speech Detection* (HS) and *Hate Speech and Offensive Language Detection* (HSO) tasks, *Coarse-Grained Stereotype Detection* (C-ST) neighboring task yields the best improvements to 71.1 and 67.39 macro-F1, respectively. All the other three neighboring tasks are useful in improving the performance of the base classifier from 66.47 and 66.13 F1 scored. For *Misogyny Detection* (MG) task, HSO neighboring task results in an improvement from 74.16 to 75.87, while the other two tasks deteriorate the performance on the task. It is also interesting to note that, the combined performance on the task with all three datasets is lower (74.78) than when using HSO data alone. For both *Coarse-* and *Fine-grained Stereotype Detection* (F-ST) tasks, HS and HSO datasets improve the performance over the baseline, while MG deteriorates the performance. The combined improvement of all the neighboring tasks together is higher than either HS

Model	Hate Speech Detection	Offense Detection	Misogyny Detection	Coarse Stereotypes	Fine Stereotypes
BERT-base	66.47	66.13	74.16	65.71	61.36
BERT-large	67.05	63.90	72.13	59.63	55.42
BART-large	68.91	65.86	73.12	63.40	54.64
XlNet-large	59.14	48.33	63.16	63.71	53.80
Multi-Task Learning					
BERT-base + MTL	69.21†	68.57†	73.48	68.29†	65.00†
BERT-large + MTL	69.78†	65.14†	73.94†	61.96†	61.65†
BART-large + MTL	67.79	68.03†	74.40†	65.77†	64.90†
XlNet-large + MTL	61.68†	46.35	64.42†	65.21†	57.00†
RL-guided MTL					
BERT-base + RL-MTL	72.06†	68.97	74.78†	74.18†	65.72†
BERT-large + RL-MTL	69.82	65.97†	75.21†	70.88†	64.74†
BART-large + RL-MTL	69.60†	66.76	75.14†	74.11†	67.94†
XlNet-large + RL-MTL	61.97	47.60†	63.21	67.98†	56.37

Table 3: Results on all the Datasets for various phases. Macro-F1 score has been reported. † indicates that improvements over the corresponding model in the previous section are statistically significant according to McNemar's statistical significance test.

T \ N	HS	HSO	MG	C-ST
HS	-	69.69	70.07	71.10
HSO	66.71	-	66.56	67.39
MG	70.98	75.87	-	73.89
C-ST	66.15	67.40	63.82	-
F-ST	63.80	63.65	59.94	56.12

Table 4: Macro-F1 scores on each Target Task in Task Ablation Study for each individual Neighbor Task. T: *Target Task*, N: *Neighboring Task*, HS: *Hate Speech Detection*, HSO: *Hate Speech and Offensive Language Detection*, MG: Misogyny Detection, C-ST: *Coarse-Grained Stereotype Detection*, F-ST: *Fine-Grained Stereotype Detection*

or HSO neighboring tasks alone. It is also interesting to note that the C-ST task doesn't contribute significantly to performance improvement on F-ST task. This might be due to the presence of anti-stereotypes and several other issues pointed out in Blodgett et al. (2021b).

8 Conclusion

We tackle the problem of *Stereotype Detection* from *data annotation* and *low-resource computational framework* perspectives in this paper. First, we discuss the key challenges that make the task unique and a low-resource one. Then, we devise a focused annotation task in conjunction with selected data candidate collection to create a fine-grained evaluation set for the task.

Further, we utilize several neighboring tasks that are correlated with our target task of *'Stereotype Detection'*, with an abundance of high-quality gold data. We propose a reinforcement learning-guided multitask learning framework that learns to select relevant examples from the neighboring tasks that improve performance on the target task. Finally, we perform exhaustive empirical experiments to showcase the effectiveness of the framework and delve into various details of the learning process via several ablation studies.

Acknowledgments

We thank the anonymous reviewers and meta-reviewer for their insightful comments that helped in improving our paper.

References

Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *The World Wide Web Conference*, pages 49–59.

Richard A. Berk. 2017. An impact assessment of machine learning risk forecasts on parole board decisions and recidivism. *Journal of Experimental Criminology*, 13:193–216.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021a. Stereotyp-

ing norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. In *ACL-IJCNLP 2021*.

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021b. Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. 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 1004–1015, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv*.

Mike Cardwell. 1996. *Dictionary of psychology*. Routledge.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Alexander Czopp. 2008. When is a compliment not a compliment? evaluating expressions of positive stereotypes. *Journal of Experimental Social Psychology*, 44:413–420.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

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.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China. Association for Computational Linguistics.

Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. In *IberEval@ SEPLN*, pages 214–228.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Kathleen C. Fraser, Isar Nejadgholi, and Svetlana Kiritchenko. 2021. Understanding and countering stereotypes: A computational approach to the stereotype content model. 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 600–616, Online. Association for Computational Linguistics.

Vijay Konda and John Tsitsiklis. 2000. Actor-critic algorithms. In *SIAM Journal on Control and Optimization*, pages 1008–1014. MIT Press.

Ksenia Konyushkova, Konrad Zolna, Yusuf Aytar, Alexander Novikov, Scott Reed, Serkan Cabi, and Nando de Freitas. 2020. Semi-supervised reward learning for offline reinforcement learning. *arXiv preprint arXiv:2012.06899*.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. 2020. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Tackling online abuse: A survey of automated abuse detection methods. *CoRR*, abs/1908.06024.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv*.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. Crows-pairs: A challenge dataset for measuring social biases in masked language models. *arXiv*.

Alexandra Olteanu, Fernando Diaz, and Gabriella Kazai. 2020. When are search completion suggestions problematic? *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–25.

Pooja Parekh and Hetal Patel. 2017. Toxic comment tools: A case study. *International Journal of Advanced Research in Computer Science*, 8(5).

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Shruti Phadke and Tanushree Mitra. 2020. Many faced hate: A cross platform study of content framing and information sharing by online hate groups. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Elena Shushkevich and John Cardiff. 2019. Automatic misogyny detection in social media: A survey. *Computación y Sistemas*, 23.

Brendan Szendro. 2021. Suicide, social capital, and hate groups in the united states. *World Affairs*, page 00438200211053889.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

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.

Yongxin Yang and Timothy M. Hospedales. 2016. Trace norm regularised deep multi-task learning. *CoRR*, abs/1606.04038.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, SuHang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024, Online. Association for Computational Linguistics.