Noname manuscript No.
(will be inserted by the editor)

Bridging the Gaps

Multi-Task Learning for Domain Transfer of Hate Speech Detection

Zeerak Waseem · James Thorne · Joachim Bingel

Received: date / Accepted: date

Abstract Accurately detecting hate speech using supervised classification is dependent on data that is annotated by humans. Attaining high agreement amongst annotators though is difficult due to the subjective nature of the task, and different cultural, geographic and social backgrounds of the annotators. Furthermore, existing datasets capture only single types of hate speech such as sexism or racism; or single demographics such as people living in the United States, which negatively affects the recall when classifying data that are not captured in the training examples. End users of websites where hate speech may occur are exposed to risk of being exposed to explicit content due to the shortcomings in the training of automatic hate speech detection systems where unseen forms of hate speech or hate speech towards unseen groups are not captured.

In this paper, we investigate methods for bridging differences in annotation and data collection of abusive language tweets such as different annotation schemes, labels, or geographic and cultural influences from data sampling. We consider three distinct sets of annotations, namely the annotations provided by [43], [45], and [16]. Specifically, we train a machine learning model using a multi-task learning (MTL) framework, where typically some auxiliary task is learned alongside a main task in order to gain better performance on the latter. Our approach distinguishes itself from most previous work in that we aim to train a model that is robust across data originating from different distributions and labeled under differing annotation guidelines, and that we understand these different datasets as different learning

All authors contributed equally.

Z. Waseem
University of Sheffield E-mail: z.w.butt@sheffield.ac.uk

J. Thorne
University of Sheffield E-mail: j.thorne@sheffield.ac.uk

J. Bingel
University of Copenhagen, E-mail: bingel@di.ku.dk

objectives in the way that classical work in multi-task learning does with different tasks.

Here, we experiment with using fine-grained tags for annotation. Aided by the predictions in our models as well as the baseline models, we seek to show that it is possible to utilize distinct domains for classification as well as showing how cultural contexts influence classifier performance as the datasets we use are collected either exclusively from the U.S. [16] or collected globally with no geographic restriction [43, 45].

Our choice for a multi-task learning set-up is motivated by a number of factors. Most importantly, MTL allows us to share knowledge between two or more objectives, such that we can leverage information encoded in one dataset to better fit another. As shown by [3] and [30], this is particularly promising when the auxiliary task has a more coarse-grained set of labels in comparison to the main task. Another benefit of MTL is that it lets us learn lower-level representations from greater amounts of data when compared to a single-task setup. This, in connection with MTL being known to work as a regularizer, is not only promising when it comes to fitting the training data, but also helps to prevent overfitting, especially when we have to deal with small datasets.

Keywords Multi-Task Learning · Abusive Language Detection · Social Media Analysis · Domain Transfer

1 Introduction

With the growing amount of user-generated content online, issues such as online abuse become more important to tackle as they affect a great number of people. A recent study undertaken by the Pew Research Center found that 73% of online adult users had witnessed online harassment, and 40% had been personally targeted [35]. Given staggering numbers such as these, it is clear that current methods of detecting abuse deployed on internet platforms are not effective in shielding users from witnessing or experiencing these forms of violence or harassment.

Alongside the pressure generated by public outcry [13], multiple government agencies have applied political pressure on social media companies to tackle the threat of online hate speech and abuse [42]. For example: the British Home Office created an action plan to deal with hate crime, in which online hate speech is explicitly mentioned [23]; Germany has introduced a €50 million fine for social media companies systematically failing to remove hate speech within 24 hours [42]; and the European Commission has set released a code of conduct for dealing with hate speech online [19].

As it stands, a vast majority of the moderation of abusive language and hate speech for these platforms is performed by human moderators, in spite of the exposure to online abuse having a profound impact on mental wellbeing [28, 40, 6]. Notably, two previous moderators have sued Microsoft for negligent infliction of emotional distress resulting in post traumatic stress disorder for being tasked with moderating child abuse [28]. Studies have shown the adverse effects of cyberbullying on youth [40] and negative effects on self-esteem of adults that were exposed to online hate speech [6]. Exposing moderation staff to every abusive post reported on an online platform has the potential to cause harm. Not only is it possible to mitigate this risk through detection of the explicit materials using automated means,

automatically detecting and auto-moderating content containing hate speech will limit exposure of offensive materials to the end users, thereby reducing risk the negative consequences. For example, it could be possible to shield children from cyberbullying.

Furthermore, correlations between increases in hate speech online and increases in hate crime have been shown [32]. Thus, not only are online safety and mental health compromised by hate speech, but offline safety can be ensured by being able to detect such increases and alerting authorities of potential risks of increase in hate crime.

1.1 Hate speech detection

Considering the task of detecting hate speech, it is important to recall that word senses may change as the dialect, sociolect, language, and culture changes [36, 8]. Currently, computational methods for hate speech detection cannot and do not try to consider the influence of socio-demographic variables. Furthermore, the issue of cultural and sociodemographic influences on the data sample are not considered, nor has the consideration of how to overcome these cultural differences in datasets collection.

The influence of these issues culminate in models that are guaranteed to have poor generalization when they are applied to different socio-demographic or cultural contexts. For example, a model trained on Standard American English (SAE) on a Twitter data sample will be likely to evaluate the use of the *n-word* as being offensive when applied to a sample of tweets that are written in African American Vernacular English (AAVE) in spite of the cultural context and acceptability of the use of the word being completely different as it is likely to primarily be African Americans writing in this dialect.

A further issue that affects generalization is the data sampling and annotation methodology. When considering data samples that are collected from two similar but different cultures, hate speech directed to one particular demographic may contain targeted locutions that may only appear offensive to one community. Distinct sets of annotation guidelines may be generated that are specific to the task [43, 45, 16, 46]. This in turn increases the barrier for combining the datasets and using them to train models on hate speech that can detect multiple forms of abuse.

Finally, given a number of datasets sampled from distinct cultural contexts, a possible approach for inducing a joint model from these might be to concatenate the datasets.[1] However, differences in size between these datasets may lead to a bias for the larger dataset, and by extension a bias for the culture captured within it. In this case, the detection of hate speech would be biased towards the cultural assumptions that this dataset makes. In contrast to simply merging datasets, multi-task learning allows us to differentiate between them while still training a common model that exploits their commonalities. Careful optimization of hyperparameters, e.g. pertaining to model topology or differing learning rates for the individual datasets, further allows us to explicitly control and correct for a potential bias, or to introduce a certain bias if we deem this desirable. We discuss multi-task learning in more detail below.

[1] After re-annotation to unify class labels, if necessary.

1.2 Multi-task learning

In this work, we seek to address the shortcoming of the previous work by considering issues of generalizability of models to accurately classify hate speech and offensive language across datasets and cultural contexts. To tackle these problems, we make use of multi-task learning (MTL), a machine learning framework that seeks to utilize the similarities and subtle differences in annotations and datasets to improve performance on and regularize against another. To the best of our knowledge, this is the first work exploring the utility of multi-task learning for abusive language.

1.2.1 Motivation

Multi-task learning has its origins in the seminal works by Caruana [10,9] and has since been applied to a wide range of areas in machine learning, including computer vision [21], bio-informatics [37] and numerous subfields of natural language processing [27,3,30]. The core idea in multi-task learning is to train a model that generates outputs for several related tasks from a single common input. We contrast this against classical machine learning approaches where typically a model is a function from one input to a single output space.

The rationale behind this idea is that certain information, which is encoded in the training data of some task, may help the model generalize better when learning how to make predictions for another related task. We can draw parallels to intuitions and observations we can make about human learning: whenever we learn a new skill, we build on other skills that we may have gained earlier. For example, when learning a foreign language, we benefit from other languages that we have learned in the past. This benefit is particularly strong when the languages in question are closely related, i.e. when they share a lot of their vocabulary or structure.[2]

From a more theoretical point of view, multi-task learning has the benefit of serving as a regularizer to a certain task which allows models to be constructed that can generalize better to unseen data. More specifically, because we simultaneously optimize parameters for several tasks, the additional information that is encoded in the auxiliary tasks acts as a mechanism which prevents the model from overfitting to the training data and becoming so specific that new data, while from the same domain and general distribution, cannot be modeled well. Previous work [3] also suggests that MTL can help a model escape from local optima, i.e. suboptimal solutions, in which it would get stuck in a single-task scenario. It has also been observed that in sequence-to-sequence architectures, the inductive bias introduced by MTL tends to have strikingly similar effects to an attention mechanism typically found in neural decoders [7], suggesting that MTL helps to focus attention on relevant parts of the input.

Another advantage of MTL that we exploit in this work is the ability to learn from multiple disjoint datasets. This means that we can combine datasets from more or less different tasks without the need for re-annotating the other data so

[2] The fact that in MTL we tend to learn both tasks simultaneously rather than in succession weakens this analogy to some degree. In fact, the simultaneous learning of two languages could actually make learning harder for humans. For a machine, however, the temporal order is less critical given its far superior memory when compared to humans.

that the label spaces are the same. This is because, as explained in Section 3.3, we can alternate between optimizing for different tasks during the training process. A consequence of this is that we can benefit from both an augmented data source while ensuring the model to generalize better across different kinds of input (e.g. tweets which originate from different domains and demographies).

1.2.2 Task choices

While the simultaneous language learning analogy above illustrates an approach to MTL that has received relatively large popularity in natural language processing, one is often only concerned with one particular task while leveraging other tasks to help this process. In such a scenario, it is common practice to distinguish between these as a *primary task* and one or more *auxiliary tasks*.

The relative importance of the tasks that we specify influences some of the design decisions of the modeling and training our data in a multi-task environment.[3] If, for example, we are ultimately only interested in a single task, we obviously only want to optimize our model architecture (and selection of auxiliary tasks) to yield the best possible performance for that primary task. If, however, we are equally interested in good performance across all tasks, our job becomes considerably harder, as we potentially need to find a compromise between performance scores across all tasks.

As discussed in [4], there are two distinct approaches to choosing an auxiliary task in the language processing architecture. The first is to select one or several tasks that are similar in their linguistic annotations to the main task (e.g. to induce better dependency parsing models by also letting the model learn syntactic categories such as parts-of-speech). The second approach is to use some non-linguistic auxiliary task whose annotations encode some signal that could be useful for the main task. A particularly interesting example is that of [27], where eye-tracking data is used to inform sentence compression.

1.2.3 Limits of multi-task learning

Multi-task learning may not always be beneficial in improving the accuracy of a classifier. Besides increasing model complexity and training time, the relation between the tasks and the respective datasets are critical for the success of MTL. Previous studies [30, 3, 5] have shown systematically that MTL may lead to detrimental performance on the main task compared to training a single-task model, and have explored the conditions under which some task may aid another. While those findings are not always compatible, a common denominator of these studies is that a high entropy in the label distribution of the auxiliary task is beneficial for the main task. In other words, if the auxiliary task has very predictable labels, performance gains on the main task become less likely.

[3] Such choices include the number and width of the hidden layers, input representations, task-specific learning rates, training schedules, among others.

1.3 Utility of multi-task learning for hate speech detection

Training a classifier to detect hate speech in a supervised setting requires training data that has been annotated by humans. Currently available resources (e.g. [43, 45, 16]) only capture types of hate speech, or single geographies meaning that a system to detect hate speech based on these data may not correctly identify hate speech outside of this domain. Generating new training data is expensive and exposes the annotator to explicit content. By applying a multi-task learning framework, we aim to provide a method which can easily be extended and allow for generalization onto unseen forms and targets of hate speech minimizing the cost of generating new datasets.

Considering classification confidence, our approach may be used for a automated content approval system which relies on detecting multiple forms of hate speech and abuse. In such a system, documents which are predicted to be hate speech with a high confidence may be automatically rejected, whereas comments for which the prediction has low confidence may be subject to human moderation. In this way, such a system would allow for human moderators to focus on borderline cases where human cognition and ability to consider context is required, exposing the moderators to explicit materials only when absolutely necessary.

2 Data

In this work, we utilize three previously published datasets for hate speech detection on Twitter data [43, 45, 16]. As [43] and [45] are annotated using the same definition of hate speech and are in fact partially overlapping datasets, we collapse these into a single composite dataset. Below, we give a comprehensive comparison of the three datasets and their annotation methods.

Intersectionality Before we begin with our introduction to the datasets, it is important that we define a key concept: "intersectionality". Intersectionality was originally coined by [14] to describe how multiple forms of oppression may intersect and create new forms of oppression that draw on the intersecting oppressions. One important note is that being on the intersection of several forms oppression is multiplicative of the separate forms of oppression, not additive. This is seen for instance in the near invisibility of the deaths of black women perpetrated by law enforcement contrasted with the deaths of black men at the hands of law enforcement [15].

2.1 Understandings of "Hate Speech"

In this work, we make use of existing definitions of hate speech and offensive language and do not introduce or modify the definitions of these concepts. Rather, we provide a discussion of the annotation methods and the definitions used in the previously published datasets.

The definition of hate speech proposed in [45] (and subsequently in [43]) is an 11-point test whereby a tweet is classified as hate speech if any one of the

test conditions (provided in Figure 1) are met. This test is based on work in the fields of Gender Studies and Critical Race Theory (CRT). Specifically, [45] draw on the work of [31] and [14] to create their test. While intersectionality is not explicitly considered in [45], it is specifically addressed in [43] through the selection of intersectional feminists annotators. In addition, in [43], annotators are asked to select between "racism", "sexism", "neither", and "both" while [45] do not annotate for "both".

Fig. 1 11-point test for hate speech provided by Waseem and Hovy.

A tweet is offensive if it

1. uses a sexist or racial slur.
2. attacks a minority.
3. seeks to silence a minority.
4. criticizes a minority (without a well founded argument).
5. promotes, but does not directly use, hate speech or violent crime.
6. criticizes a minority and uses a straw man argument.
7. blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims.
8. shows support of problematic hash tags. E.g. "#BanIslam", "#whoriental", "#whitegenocide"
9. negatively stereotypes a minority.
10. defends xenophobia or sexism.
11. contains a screen name that is offensive, as per the previous criteria, the tweet is ambiguous (at best), and the tweet is on a topic that satisfies any of the above criteria.

In the criteria for a tweet to be annotated as hate speech in [45] and [43], we observe there are three different groups of tests (see Table 1).

Table 1 Types of hate speech in annotation guidelines of [45,43].

Group description	**Test numbers**
Overt aggression	1, 2, 9, 11
Defense/Support of hate speech	5, 8, 10
Subversive aggression	3, 4, 6, 7

Considering the guidelines presented in Figure 1 and the categorization in Table 1 in further detail, it is apparent that the aim of these guidelines was to

capture a broad spectrum of the hostile experiences that oppressed groups in society face. We can visualize a quadrant describing the types of abuse these guidelines capture. Along one axis, abuse can range from explicit to implicit. And the second axis, abuse can range from directed to generalized [44]. These two datasets [43,45] attempt to capture both explicit and implicit hate speech that can be either directed or generalized.

In consideration of hate speech, offensive language, and more generally abusive language, it is important to note that the use of slurs and profanity may not be indicators of abuse. For instance [36] argues that while the *n-word* is considered an offensive term in many contexts, it is not an offensive term when used within the African American community, instead it can function as a way of communicating solidarity and framing oneself within the historical context of the oppression of African Americans in the United States, and as [36] writes:

> "Using nigga to address and refer can contribute to the construction of a speakers identity, but as in the segment above, it can also ascribe identity (Coupland 2007) to a referent or addressee as a coparticipant in the diaspora."

Waseem and Hovy's annotation method does not explicitly afford context dependent annotation, seen through test 1. As such, any use of the *n-word* may be annotated as hate speech.

In comparison, [16] employ a different definition, in which they move away from the categories of sexism and racism and employ the term "target group", which suggests a move away from the literature in Gender studies and CRT. Further, they more clearly move away from the literature by basing their definition of hate speech in the user guidelines of Facebook and Twitter. Thus, they reach a definition that erases the societal context within which hate speech occurs and those who are most frequently targets of it:

> "language that is used to expresses hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group."

Using this definition, [16] ask their annotators to distinguish between "hate speech", "offensive", and "neither". They allow for distinguishing between these by instructing their annotators to take context in which the message was sent into account and explicitly state that the use of profanity or slurs does not necessarily indicate hate speech, it may simply be offensive depending on the context. Thus they seek to reintroduce *a context* after erasing a societal context from their definition.[4] Considering the case of the *n-word* and AAVE, this annotation method allows for it not to be tagged as hate speech:

> "Users were asked to think not just about the words appearing in a given tweet but about the context in which they were used. They were instructed that the presence of a particular word, however offensive, did not necessarily indicate a tweet is hate speech"

[4] Context is not defined more clearly in their paper.

Thus suggesting that while the use of a term may not be hate speech, it is still offensive and as such the *n-word* may still be flagged as offensive when used within the African-American community. Interestingly, [16] find that annotators tend to regard homophobic and racist language more likely to be hate speech whereas sexist language is more often flagged as offensive.

In this work, we do not distinguish between the two definitions and annotation methods as our aim is to investigate methods for domain adaptation from one datasets onto the other.

2.2 Commonalities & Differences

Here we give a brief overview of several of the commonalities and differences that are found amongst the three utilized datasets.

In [43] and [45], the same annotation guidelines are used. However, there are also key differences to be found between the two datatsets: in [45], two annotators label the datasets, whereas in [43] the datasets is annotated by a group of activist intersectional feminists and another set of annotations is obtained by crowdsourcing the annotation efforts on CrowdFlower. The annotations from [43] that we employ are the feminist annotations. In [16] the annotations are similarly crowdsourced on CrowdFlower.

All three datasets are collected from Twitter. However, while [16] collect tweets written within the United States of America, [45] and [43] do not limit by geographic location. To mitigate the different geographical (and thereby cultural) biases that arise from the different sampling of these datasets is one of the contributions we seek to make with our work

One of the key differences between the two definitions of hate speech is its positioning of within societal structures. By basing their test in Gender Studies and CRT, [45] implicitly place their work within the notion of structural inequality, By using charged terms such as "sexist and racial slur" and "attacks a minority", they explicitly frame their work within the context of abuse not being equally distributed amongst all groups.

On the other hand, [16] do not frame their work within this context nor do they base their definitions in the previous literature. Given that they base their definition in the guidelines of social media companies, it is based in law, as their guidelines are placed within the context of corporations that seek to react to a user base that highlights their discomfort on their platform while simultaneously navigating the legal realities of multiple nations. One such reality is that, within the U.S.A. anti-subordination is a complicated area to navigate, and for a corporation it is unnecessary to do so when it is possible to frame within an anti-discrimination context.

Beyond these differences, another difference occurs in the targets within the "hate speech" and "racism" classes. As previously noted, annotators were more likely to find "hate speech" to be racist or homophobic speech, while sexist speech was more likely to be "offensive" [16]. Thus considering the targets of racism between the datasets, the main targets of racism in [43] and [45] are Muslims, whereas the main targets of racism in [16] are African Americans.

Finally, the definitions, labels, and the annotation scheme differ slightly as covered in Section 2.1.

3 Model

We use a deep multi-task model to transfer knowledge between different tasks. While a number of different approaches to MTL have been explored in the past, the paradigm that has attracted most attention in deep learning and natural language processing (NLP) in particular is *hard parameter sharing*. As its name suggests, this MTL paradigm works by sharing a subset of a model's parameters between different tasks. From a different but equivalent perspective, this is building distinct models for each task, with these models sharing (and jointly optimizing) some of their parameters.

We will compare the performance of these MTL models against simple baseline models without hard parameter sharing. We first introduce and define the multi-layer perceptron feed forward neural network and then discuss modifications that allow hard sharing of parameters

3.1 Baseline Model Definition

We build a very simple feed-forward neural network without any parameter sharing. This model which takes as its input some fixed-size representation x of a tweet and computes a hidden latent representation h, which is a linear projection of x using a matrix of weights W_0 and a bias term b_0, followed by a non-linear transformation:

$$h_0 = \tanh(xW_0 + b_0) \tag{1}$$

For a deeper model, further hidden representations h_l are computed accordingly by stacking these layers. The respective previous hidden layer output h_{l-1} is provided as the inputs to following layer.

$$h_l = \tanh(h_{l-1}W_l + b_l) \tag{2}$$

The final hidden representation h_L is then used to compute the model output:

$$y = \sigma(h_L W_{out} + b_{out}) \tag{3}$$

Typically, σ is the softmax function which, for a k-dimensional input, normalizes the output to the range $[0, 1]$ such that its sum is 1, representing a categorical distribution over outputs.

$$\sigma(z) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{4}$$

3.2 Multi-task Model Definition

In Equation 3, the model uses the parameters W_{out} and b_{out} to predict outputs for a single classification task. This model can be extended to predict outputs for more than one task, with the use of hard parameter sharing through the introduction of additional parameters that are specific for each task t:

$$y_t = \sigma(h_L W_t + b_t) \tag{5}$$

In this setup, the weights W_l and bias terms b_l are thus the parameters that are shared between the tasks and learned jointly. In constrast, the weights W_t and bias terms b_t at the output layer are specific to only one task.[5]

3.3 Training

Our goal is to learn parameters for the model in a supervised learning scenario using labeled training data. We assume that training data is provided as set labeled pairs of instances x_i and labels y_i: $\{(x_i, y_i)\}_{i=1}^{N}$. Training the neural network is the process of optimizing the parameters with the objective of reducing the error rate of predicted outputs $\hat{y}_i$ with respect to the annotated labels y_i.

The common way to optimize a deep learning model is via the so-called forward-backward algorithm, where we first compute some guess that the model produces for a given input example (following Eq. 5) and then compare this to the ground truth that is annotated in our training data. This comparison is a quantity that measures the error (or *loss*), which we then use to inform our model how strongly it should change its parameters during the backward pass in order to arrive at a better guess in the next iteration. This is typically done using some flavor of gradient descent.

A challenge that multi-task learning now poses in comparison to a classical single-task scenario is that we do not have a single loss that we can use to optimize our model, but one for every task. This raises the question of how to schedule the training across the different tasks. A common technique is to flip a coin at every training iteration (i.e. for every forward-backward pass) that decides for which task we are going to train. Depending on the outcome, we then sample a batch of training data for this task and optimize the parameters that are involved in predicting the respective output for this task. An obvious alternative to flipping a coin would be a strictly defined schedule that alternates between the tasks at every iteration, or a biased coin or schedule that gives preference to some tasks over others.

In this work, we select the coin-flipping strategy and sample training data for a task that we choose randomly with equal probability. We also follow standard practice in employing a dropout regularizer on each hidden layer during training, where we randomly set units in the hidden representations computed in Eq. 1 to zero with 0.2 probability.

[5] Note that in principle, hard parameter sharing also allows us to predict the different tasks at different depths of the model, e.g. to compute the output for task A from some hidden representation h_m and task B from h_n (with $m \neq n$). Yet another possible variation is to compute further hidden representations that are task-specific and not shared, but ultimately draw on some common lower-level representation.

3.4 Features

Our model utilizes fixed-size representations of the input (thereby differing from more complex deep learning architectures like recurrent or convolutional neural networks). We use two classes of features representation: (1) a Bag-of-Words representation of tweet words, bigrams and character n-grams, and (2) continuous word representations. We perform an evaluation of both in isolation as well as a combination of the two.

3.4.1 Bag-of-Words (BoW) Representation of Features

We construct a vocabulary of words occurring within our corpus of tweets and restrict our Bag-of-Words representation to the 5000 most frequently occurring words to prevent our model from overfitting the Zipf long tail. Each occurrence of a word is modeled as a one-hot vector that is summed for each tweet.[6]

In addition, we collect word pairs (bigrams) and concatenate these into a single word and also add the most frequent 5000 to our vocabulary. For example: "go away" would be added to the vocabulary as the token "go_away". The intuition behind this is that multi-word expressions that have a meaning that is distinct from their constituent words can be more accurately represented as its their own tokens rather than having the meaning diluted by other training examples.

As we are expecting to classify less formal and non-standard uses of English on Twitter, we must also account for word mis-spellings, alterations and colloquial style. For example: the words "yeah", "ye", "yep", "yea" and "yes" all convey similar meaning but would be represented as five distinct tokens using a Bag-of-Words model. We account for this through character segmentation as well as word segmentation. The segment "ye" appears 5 times and (in conjunction with other observed features) convey part of the meaning. We extract character bigrams (character pairs) and character trigrams (groups of three letters) and treat these as words in our vocabulary. Again, we only use the 5000 most common in our vocabulary.

3.4.2 Sub-word Embeddings

Rather than encoding the meaning of a word as a single one-hot vector that is the size of the vocabulary, word embeddings represent the meaning of tokens as low-dimensional real-valued vector. Typically, this vector may be between 100 to 300 dimensions. These dense meaning representations can be designed such that words which convey similar meanings or appear in similar contexts also have similar vector-based representations.

We choose to use embeddings because this allows us to capture from different, but related concepts that occur in our data that cannot so easily be represented with the symbolic Bag-of-Words representation. For example, the encoding of a tweet containing the word 'football' will be entirely different from the a tweet containing the word 'soccer' using a BoW representation - even though these are similar concepts. Using continuous representations enables some cross talk between

[6] A one-hot vector is a binary vector of indicator features that are 1 if that feature occurs in the document otherwise 0 in the feature does not occur in document.

different concepts encountered during training. We hypothesize this may yield a classifier that is less prone to over-fitting the distribution of data that it observed during training and more accurate for unseen out of domain data.

Our multi-layer perceptron models are designed with a fixed input representation size. However, the size of tweets is variable. To train our model, we must make a fixed size representation of a variable size input. While it is normal in text classification problems perform convolutions [26] over the input data or to train train a time-series model such as a Recurrent Neural Network, the limited size of the training data available for this task prevents use from using these techniques. Instead we a perform a pooling operation by averaging all the vectors in the tweet which is shown to yield an acceptable (yet suboptimal) performance on other text classification tasks [41].

Because language on Twitter is informal, we expect to encounter unseen words and variations of known words. Rather than using word representations, we use vector representations of sub-word units similar to morphemes [22] which will allow us to better capture common word units that are occurring in this informal language.

3.5 Pre-processing

We pre-process all tweets with the following steps: usernames and mentions are converted to a single type to aid anonymity and to also prevent bias in the training that may occur by learning associations between usernames rather than language. URLs and Hashtags are filtered out for the same reason. Furthermore, we convert all text to lower case and normalize numbers to a special digit symbol. Finally, all line breaks in tweets are replaced with spaces.

4 Experiments

To test our hypothesis, we construct three experimental configurations which we test out using our models. For our configurations we use the same two datasets described in Section 2, namely the composition of the datasets from [45] and [43], and the dataset from [16]. For all three configurations we test our models using each dataset as the training dataset in turn. Further, we conduct three different experiments with different features for each configuration, a lexical model using BoW, a model using only embeddings, and a model using both BoW and embeddings. In each case, we train our models on a total of 45 iterations over the available training data and finally test it using the parameters which yield the best performance on the held-out development set, preventing overfitting on the training data.

4.1 Baseline models

We construct our baseline models using by training a model on a single datasets and predicting on another as has been attempted in previous work [43]. We select this as our baseline as it has been been attempted in previous work with low

success and therefore will highlight the issues with attempting to predict on one dataset given that a model is trained on another. Consistent with previous work, we expect these models to have poor performance on out of domain data. By using each dataset in turn for training and predicting on the other, we show that it is not simply a questions of which dataset our models are trained on but rather that regardless of which dataset we train on, the capabilities of a model to predict on a dataset which is collected in a different culture, with different ways of using language, and with different targets and topics will be poor unless we specifically seek to address this.

To evaluate the performance of the classifier on the out of domain data, we defined a deterministic class mapping between the two datasets based on observations. We map the "Neither" class from [43, 45] to the "Not Offensive" class in [16]. We also observe that in [16], a large majority of the tweets annotated as offensive language are sexist so we map the "Offensive" class to "Sexist". A large majority of the tweets labeled as hate-speech contain racist slurs and remarks, so we map the "Hate Speech" in [16] class to the "Racism" class in [43, 45].

4.2 Composite data models

In this configuration, we build a composite of all three datasets into a single training set and test set. With this model we seek to build a strong baseline as we expect this will outperform the simple baseline models and simultaneously will allow for us to test the performance of a model where a composition of all known datasets is performed. Additionally, this method allows us to test whether the influence of using a multi-task learning configuration only shows benefits due to the model being exposed to all available datasets. Finally, using the composite datasets we test whether the distribution of documents from each dataset influences which evaluation dataset the model performs best on.

4.3 Multi-task learning models

Our third configuration uses a multi-task learning framework, in which we test for whether simultaneously learning to predict on two different datasets with a shared representation can outperform a strong baseline. Furthermore, by utilizing this approach, we test the potential of domain transfer for abusive language via multi-tasking. Finally, this setup allows us to test whether cultural influences and differences can be utilized such that prediction is improved on a dataset whose collection is based in a different cultural context. Given that it differentiates between the primary and auxiliary tasks while still learning from both datasets, we expect this configuration to outperform the other two model types. Specifically, we expect it to outperform our simple baseline models by a large margin and, to a lesser degree, our composite data models.

We test two conditions for this set of experiments, alternating between which of the two datasets we use as a main task, with the other serving as the auxiliary training data. While our model internally treats both tasks equally, the difference between these two scenarios is that we only tune the model on the development set of the respective main task.

4.4 Dataset statistics

We construct a dataset for "racism"/"sexism" detection by merging the [43] and [45] datasets. In [45], only the classes "racism", "sexism", and "neither" are utilized, however due to the focus on intersectional abuse [43] also annotate for "both". In our experiments we augment the "racism" class with documents labeled as "both" as there are only 49 documents labeled as "both" and because [16] is not annotated for the intersections but rather "hate speech" and "offensive." Therefore, training to detect this composite class, regardless of which dataset is trained on or is used as the primary task, would hardly be successful. Furthermore, the issues with detecting the class would become greater as we create splits in our dataset for training and testing purposes. Finally, we augment the "racism" class with the documents from the "both" class, as "racism" is the class with the fewest documents, and as such increasing the number of documents is more likely to improve performance on the class rather than the "sexism" class which has more documents, even if the increase in number of documents is negligible.

To build our model, we create stratified splits of our dataset to ensure that class balance across different splits remains the same. We generate a split for training our model, a split for development evaluation, and a final evaluation (see Tables 2 and 3 for dataset statistics across the splits) dataset which is entirely unseen for our models at test time.

Table 2 Dataset statistics of the Waseem[43]/Waseem-Hovy[45] and splits produced for training, developing and evaluating the models.

Dataset Split	Racism	Sexism	Neither
Training	1697	3365	11688
Development	211	420	1461
Test	214	423	1461
Total	2122	4208	14610

Table 3 Dataset statistics of the Davidson[16] and splits produced for training, developing and evaluating the models.

Dataset Split	Hate Speech	Offensive	Not Offensive
Training	1144	15352	3330
Development	143	1919	416
Test	143	1919	417
Total	1430	19190	4163

4.5 Evaluation Metrics

Given the high imbalance between positive classes seen in Tables 2 and 3, that is the "racist", "sexist", "offensive", or "hate speech" classes, it is important that we evaluate using metrics that are not susceptible to class imbalances. For instance, a metric that would be susceptible to class imbalances is accuracy, which simply

calculates the fraction of all correct predictions over all documents in the evaluation set. Thus, if one class dominates the dataset, and a classifier performs well on that class but poorly on all other classes, the accuracy score would still show a quite high score. For this reason, we provide precision, recall, and weighted-average F1-scores for each class as well as their average. As such we can show the actual performance on our task, rather than a biased sample. Below we provide definitions and explanations of our metrics.

4.5.1 Precision, Recall, and F1-score

We compute the precision, recall, and F_1-score, and report F_1-scores, as these measures are robust against class imbalance while providing insight into the performance of our models. For all three, in the class-based representation the "positive" class refers to the class which we are predicting for.

Precision describes the fraction of how many of the examples which our model predicted to belong to one of the positive classes actually belonged to those positive classes. Thus, it provides us with a insight of how often other classes are misclassified as this class.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{6}$$

Recall, on the other hand, describes how often our models predicted the correct class as a proportion of all predictions; providing insight how often the classifier misclassifies the this class as another class.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{7}$$

The F_1-score is the harmonic mean between precision and recall which penalizes imbalance between precision and recall.

$$F_1\text{-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{8}$$

5 Experimental Results

In this section, we present the results of our experiments. Results of all experiments are presented in Table 4. Each subsection will highlight one type of model and analyze and discuss the performance of that model. We will be comparing across datasets with respect to the class distributions. Please refer to Tables 2 and 3 for class distributions.

5.1 Single-task baseline models

Our single-task baseline models are built using the same method that has been used in previous work to predict on out-of-domain data, namely training on an in-domain training set and predicting on an out-of-domain test set [43]. In this we show findings consistent with previous work, namely that in-domain prediction

Table 4 Comparison of test-set performance of within-domain and out-of-domain datasets using models trained only on one dataset (first four rows), models trained by concatenating both datasets (middle two rows), and using both datasets in a multi-task learning environment (final four rows). For each training regime, we compare using Bag-of-Words (BoW), the Average of Subword Embeddings (Emb) and both (B+E) as features for each tweet. Key: (R)acism, (S)exism, (H)ate-Speech, (O)ffensive, (N)either. Datasets: Davidson [16], Waseem[43]/Waseem-Hovy[45] (W/W+H)

Training Objective		**Feats**	**F_1-Scores of Predictions on Test Sets**							
Primary	**Aux**		**W/W+H**				**Davidson**			
			R	S	N	Avg	H	O	N	Avg
W/W+H	-	BoW	0.70	0.65	0.88	0.82	0.00	0.64	0.42	0.57
W/W+H	-	Emb	0.30	0.42	0.85	0.71	0.01	0.04	0.29	0.08
W/W+H	-	B+E	0.00	0.00	0.82	0.57	0.00	0.00	0.29	0.05
Davidson	-	BoW	0.22	0.29	0.69	0.56	0.32	0.94	0.84	0.89
Davidson	-	Emb	0.00	0.32	0.60	0.48	0.19	0.92	0.69	0.84
Davidson	-	B+E	0.25	0.33	0.70	0.58	0.39	0.82	0.94	0.89
Both	-	BoW	0.21	0.54	0.81	0.70	0.20	0.92	0.77	0.86
Both	-	Emb	0.21	0.45	0.76	0.64	0.05	0.90	0.64	0.80
Both	-	B+E	0.17	0.53	0.81	0.69	0.31	0.92	0.77	0.86
W/W+H	Davidson	BoW	0.64	0.63	0.87	0.80	0.39	0.94	0.84	0.89
W/W+H	Davidson	Emb	0.32	0.50	0.84	0.72	0.10	0.91	0.64	0.82
W/W+H	Davidson	B+E	0.51	0.53	0.86	0.75	0.16	0.93	0.78	0.86
Davidson	W/W+H	BoW	0.66	0.62	0.86	0.79	0.37	0.94	0.83	0.89
Davidson	W/W+H	Emb	0.39	0.49	0.84	0.73	0.09	0.91	0.62	0.81
Davidson	W/W+H	B+E	0.60	0.57	0.85	0.77	0.14	0.93	0.78	0.86

performs reliably when using simple features and models such as our MLP with BoW features.

Considering the results of out of domain classification presented in the first six rows in Table 4, we observe that the performance is extremely poor. While the F_1-scores for minority classes performs below chance, the *Offensive* class out of domain the F_1-scores for the majority class are, in some instances, slightly more respectable.

Considering the average performance over all classes, we observe significant drop in F_1-score from the in-domain dataset to the out-of-domain dataset. This baseline shows that performance on out-of-domain datasets will be poor regardless of which single-domain dataset is used as the training set when the datasets have different underlying distributions and label schemata.

5.2 Composite dataset models

With our composite dataset models, we sought to build a strong baseline which used both datasets to allow comparison against our multi-task learning models. In our results, we observe that the performance on the minority classes is around the level of random chance while the performance of majority class is satisfactory. Considering the average F_1 score, these models perform well compared to our single-task baseline models. However, this performance is less than desirable.

We observe that inclusion of the second dataset in training reduces average classification performance of BoW models in comparison to our in-domain baselines which only use a single dataset. While in comparison the in-domain performance

the accuracy is reduced, in comparison to the out-of-domain performance we observe a marked rise. This provides evidence to suggest that while the model will be better at "generalizing" between the multiple datasets, it will do so at the cost of in-domain performance on the distinct datasets from which it is built.

In our single-task baseline, we observe that for the Waseem/Waseem-Hovy data, the Embedding-based features yield poor classification performance. This may be due to data scarcity for the minority classes (racism and sexism). In the composite dataset, we observe an improvement in classification performance for these values when using embedding-based features due to the inclusion of the additional data.

5.3 Multi-task learning models

In all cases, the application of multi-task learning (presented in the final six rows of Table 4) yields clear improvements in the average F_1 classification performance in comparison to the composite dataset as well as to the cross-domain scenarios, outperforming our strong and weak baselines. Notably, these improvements are achieved with minimal loss of performance compared to the in-domain performance of the single task model. We observe four instances where the score was reduced: the average reduction in these cases was 0.025.

The choice of a primary versus auxiliary task appears to have little effect on either test set, which is relatively unsurprising given that the main task choice solely impacts the final model selection criterion rather than training itself.

Our results imply that the MTL approach can overcome the problems that arise from differing annotation schemes for hate speech detection stemming from cultural influences and differences. This poses the central contribution of our work and, extrapolated to a more general case, suggests that the improved generalization that comes with a multi-task learning approach can bridge gaps between different domains and annotation schemes in several other tasks. This is, to our knowledge, an application of multi-task learning that has previously received little attention and is worth exploring further.

5.4 Critiques of Datasets

Referring back to Section 2, we find one troubling aspect of the data released by [16]. While their work is interesting and profound there is a serious issue in their data which we discovered quite late in our process of writing this, and had we been aware of it at an earlier stage we would not have used their datasets. The issue that we found is that a large part of their positive classes consist of African American Vernacular English, and while we encourage research to work on abusive language and AAVE, the combination should be handled with care. As a large majority of the datasets is written in AAVE, we consider the use of the *n-word*. The *n-word* occurs with a 'ga' ending 2167 times. It is labeled as either "offensive" or "hate speech" a total of 2161 times. This includes examples such as:[7] "This Niggah Kevin Hart couldn't sit down lmaooooooooooo My niggah My

[7] Emoticons used in the text are removed, urls are replaced with "<url>" token, and usernames are replaced with "@user".

Niggah", "If I wanted my ex back believe me I'd fucking go get they ass. but I ain't bout to dig through the trash.", and "@user Police just tried to Rodney King a nigga... happen to my nig out here". Considering these examples within the frame of AAVE, it is clear that these are not offensive, nor do they appear to contain other signals of abuse or offensive language, yet they were all labeled as "offensive". We determine that these tweets are in fact AAVE using the references to African American celebrities,[8] the use of phonologically motivated spelling variations and contractions [25], and the reference to the police brutality, including the fact that not only is the user describing the threat of police brutality to themselves, but also referring to someone they know who has been a victim of police brutality from which we illicit is AAVE due to the over-policing of black communities [17]. Considering these factors, some of which common to large sets of the dataset it becomes clear that these examples, as so many other in the dataset, are AAVE.

By training models to detect offensive language and hate speech using this dataset, researchers are implicitly also passing judgment on what is deemed acceptable sociolects and dialects. To seek to control the dialect spoken by communities that are marginalized through over-policing [17], mass incarcerated [38] and under represented in academia [1], media [18], and leadership positions [12] is callous at best and malicious at worst. While it is our contention that this dataset in its current state should not be used in terms of abusive language detection research without re-annotation, we encourage a re-annotation of this resource as it can be a valuable resource into the nature of abusive and offensive language within African American communities. Furthermore, it goes to highlight the argument in [43], that the identities of annotators is important. We find it unlikely that people from marginalized African-American communities would annotate the examples above, or the many other instances in the dataset as offensive or hate speech. Therefore, we encourage for a re-annotation with members of marginalized African American communities as the primary annotators.

We acknowledge that through their instruction for annotators to consider context, the fact that AAVE is so frequently annotated as either "hate speech" or "offensive" directly goes against the intentions of [16] and the instructions they provided their annotators with. This further highlights the importance of selecting the correct annotators for tasks such as abusive language detection.

6 Related Work

6.1 Abusive Language

Abusive language research has seen a recent increase in attention from researchers in NLP [44, 43, 16, 24, 11, 46, 33, 39] yet the focus of bridging across geographical context, cultural context, or dataset has to our knowledge only been addressed by [43], [11], and [33]. In [43], they collapse their annotations and the annotations from [45] into a "hate speech" and "not hate speech" classes, and train on [43] and predict on [45]. In [33], they build models using a mixture of lexical features and word embeddings, additionally, they split their dataset up by when the documents

[8] "My n*ggah my n*ggah" is a reference to Denzel Washington's character in the movie Training Day.

were posted and find that by adding data as the model learns improves the performance of the model. Finally, [11] take a very different approach by specifically aiming to make their models function on new datasets. Rather than assigning focus on individual documents, as with a BoW model, [11] choose to approach their task by considering multiple communities, some of which are known to be abusive. Given these abusive and non-abusive communities, they compute the distance of a comment to the communities using a BoW representation of the comment. Using this approach, [11] find that their model outperforms models that are trained within domain using lexical features. One important distinction between [11] and this work, is that [11] requires multiple distinct data sources to perform well. As shown through our use of two datasets, we do not require multiple data sources to obtain generalization.

Other work in the field has dealt with using neural networks for predicting hate speech [20,34,2]. In all three papers, they experiment with Convolutional Neural Networks (CNN). In [34] they model the task slightly differently from other previous works by first building a model to detect whether something is hate speech and then classifying it into the specific form of hate speech ([34] consider "racism", "sexism", and "neither").

6.2 Multi-task learning

As noted above, this is to our knowledge the first work that employs multi-task learning strategies to tackle hate speech detection, aiming at transferring knowledge between domains and differently annotated datasets.

An example of previous work that has used multi-task learning to build models to work well across domains is [47], where sentence representations are learned through auxiliary tasks in order to improve cross-domain sentiment classification. However, this approach critically differs from ours in that the authors do not perceive different annotations as different tasks, but create synthetic data for their auxiliary tasks that are exclusively used to learn better representations of the input.

Another interesting case is that of [29], which proposes a multi-input and multi-output sequence-to-sequence model for neural machine translation that can handle different source and target languages, encoding input from any language into the same language-independent intermediate representation, from which they decode into any available target language. While fundamentally different in model architecture and learning problem, this work shares our idea of perceiving heterogeneous datasets from different 'domains' as separate tasks to build a robust cross-domain model.

7 Conclusion

In this work, we applied the use of multi-task learning to develop classifiers for hate speech and abusive language. We find that utilizing an MTL framework for detecting hate speech allows for vastly improving the ability of a hate speech detection model to generalize to new datasets and distributions. In this work, we specifically chose datasets which were collected with distinct cultural groundings

and bias to examine the utility of MTL to overcome such biases. With this in mind, we show that MTL does in fact allow for generalization onto a different cultural context. A particular strength of our MTL approach is that its better generalization allows for a more robust application to completely novel data. In such a scenario, the outputs from the MTL model could act as a mixture of experts that jointly vote on new data. Prior knowledge could be easily integrated here in giving more weight to the sub-model whose training data we believe is closest to our new data.

Our results further show that MTL allows for comparable results to using single task models that predict in-domain, while also allowing for prediction on other datasets. Additionally, we find that a high performance model can be built using composite datasets however, MTL allows for overall improvements over it. Furthermore, we find that in our experiments the choice of primary and auxiliary task had little influence on the performance of the model. We show that applying MTL to classify hate speech on out of domain data is a vast improvement over single-task models and has a slight average improvement over the composite dataset models.

In a more practical sense, our approach simplifies the construction of broad-domain filters for moderation of content by a classifier to learn from examples from multiple different domains and tasks. This minimizes the barrier of entry for detecting hate speech from different domains and communities and thus mitigates the risk of exposing users to previously unseen forms of online hate speech and abuse. By using the confidence from scores from this approach to only expose moderators to borderline content where absolutely necessary, we can reduce the volume of explicit materials that staff members are exposed to which has the potential to reduce harm.

In conclusion, while our method does not guarantee improvements on in-domain prediction of single-task models, we introduce the use of a method that can allow for lower barriers to training and detecting new forms of hate speech and abusive language. Considering the correlation between online hate speech and hate crime, lowering entry barriers for hate speech and abusive language detection may allow for platforms to more easily protect their users from undue harm online and offline.

8 Future work

Our work raises a number of questions on how to deal with domain adaptation and abusive language. First and foremost, future work should seek to address making improvements on the minority classes. Second, this paper explores multi-task learning for domain adaptation, it could therefore be beneficial to consider other methods for domain adaptation. Additionally, future work could seek to address the use of user information and the use of demographic variables such as age, gender, and income as additional signals for detection of abusive language and hate speech across datasets. As far as our multi-task approach is concerned, future work may investigate relationships between the datasets and how they reflect in optimal hyper-parameters for the network architecture and training. For example, specific task combinations could benefit from a more fine-tuned training schedule or learning rate ratio, or the integration of further task-specific hidden layers.

References

1. Allen, W.R., Epps, E.G., Guillory, E.A., Suh, S.A., Bonous-Hammarth, M.: The black academic: Faculty status among african americans in u.s. higher education. The Journal of Negro Education **69**(1/2), 112–127 (2000). URL http://www.jstor.org/stable/2696268
2. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion, pp. 759–760. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017). DOI 10.1145/3041021.3054223. URL https://doi.org/10.1145/3041021.3054223
3. Bingel, J., Søgaard, A.: Identifying beneficial task relations for multi-task learning in deep neural networks. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 164–169. Association for Computational Linguistics, Valencia, Spain (2017). URL http://www.aclweb.org/anthology/E17-2026
4. Bjerva, J.: One model to rule them all: Multitask and multilingual modelling for lexical analysis. arXiv preprint arXiv:1711.01100 (2017)
5. Bjerva, J.: Will my auxiliary tagging task help? estimating auxiliary tasks effectivity in multi-task learning. In: Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden, 131, pp. 216–220. Linköping University Electronic Press (2017)
6. Boeckmann, R.J., Liew, J.: Hate speech: Asian american students justice judgments and psychological responses. Journal of Social Issues **58**(2), 363–381 (2002). DOI 10.1111/1540-4560.00265. URL http://dx.doi.org/10.1111/1540-4560.00265
7. Bollmann, M., Bingel, J., Søgaard, A.: Learning attention for historical text normalization by learning to pronounce. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 332–344 (2017)
8. Boyle, K.: Hate speech–the united states versus the rest of the world. Maine Law Review **53**(2), 487–502 (2001)
9. Caruana, R.: Multitask learning. In: Learning to learn, pp. 95–133. Springer (1998)
10. Caruana, R.A.: Multitask connectionist learning. In: In Proceedings of the 1993 Connectionist Models Summer School. Citeseer (1993)
11. Chandrasekharan, E., Samory, M., Srinivasan, A., Gilbert, E.: The bag of communities: Identifying abusive behavior online with preexisting internet data. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, pp. 3175–3187. ACM, New York, NY, USA (2017). DOI 10.1145/3025453.3026018. URL http://doi.acm.org/10.1145/3025453.3026018
12. Cohen, P.N., Huffman, M.L.: Black under-representation in management across u.s. labor markets. The ANNALS of the American Academy of Political and Social Science **609**(1), 181–199 (2007). DOI 10.1177/0002716206296734
13. Crawford, K., Gillespie, T.: What is a flag for? social media reporting tools and the vocabulary of complaint. New Media & Society **18**(3), 410–428 (2014). DOI 10.1177/1461444814543163. URL https://doi.org/10.1177/1461444814543163
14. Crenshaw, K.: Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist eory and antiracist politics. University of Chicago Legal Forum **1989**(1) (1989)
15. Crenshaw, K.: The urgency of intersectionality (2016). URL https://www.ted.com/talks/kimberle_crenshaw_the_urgency_of_intersectionality
16. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of ICWSM (2017)
17. Desmond-Harris, J.: Are black communities overpoliced or underpoliced? both. (2015). URL https://www.vox.com/2015/4/14/8411733/black-community-policing-crime
18. Dixon, T., Linz, D.: Overrepresentation and underrepresentation of african americans and latinos as lawbreakers on television news. Journal of Communication **50**(2), 131–154 (2000). DOI 10.1111/j.1460-2466.2000.tb02845.x. URL http://dx.doi.org/10.1111/j.1460-2466.2000.tb02845.x
19. European Commission: Code of conduct on countering illegal hate speech online. Tech. rep. (2016)
20. Gambäck, B., Sikdar, U.K.: Using convolutional neural networks to classify hate-speech. In: Proceedings of the First Workshop on Abusive Language Online, pp. 85–90. Association for Computational Linguistics (2017). URL http://aclweb.org/anthology/W17-3013

21. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440–1448 (2015)
22. Heinzerling, B., Strube, M.: Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages. CoRR **abs/1710.02187** (2017). URL http://arxiv.org/abs/1710.02187
23. Home Office: Action against hate the uk governments plan for tackling hate crime. Tech. rep. (2016)
24. Jha, A., Mamidi, R.: When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In: Proceedings of the Second Workshop on NLP and Computational Social Science, pp. 7–16. Association for Computational Linguistics (2017). URL http://aclweb.org/anthology/W17-2902
25. Jørgensen, A., Hovy, D., Søgaard, A.: Learning a pos tagger for aave-like language. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1115–1120. Association for Computational Linguistics, San Diego, California (2016). URL http://www.aclweb.org/anthology/N16-1130
26. Kim, Y.: Convolutional neural networks for sentence classification. CoRR **abs/1408.5882** (2014). URL http://arxiv.org/abs/1408.5882
27. Klerke, S., Goldberg, Y., Søgaard, A.: Improving sentence compression by learning to predict gaze. In: Proceedings of NAACL-HLT, pp. 1528–1533 (2016)
28. Levin, S.: Moderators who had to view child abuse content sue microsoft, claiming ptsd (2017)
29. Luong, M.T., Le, Q.V., Sutskever, I., Vinyals, O., Kaiser, L.: Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06114 (2015)
30. Martínez Alonso, H., Plank, B.: When is multitask learning effective? semantic sequence prediction under varying data conditions. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 44–53. Association for Computational Linguistics, Valencia, Spain (2017). URL http://www.aclweb.org/anthology/E17-1005
31. McIntosh, P.: White privilege and male privilege: A personal account of coming to see correpondences through work in women's studies (1988)
32. Müller, K., Schwarz, C.: Fanning the flames of hate: Social media and hate crime (2017)
33. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, WWW '16, pp. 145–153. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2016). DOI 10.1145/2872427.2883062. URL http://dx.doi.org/10.1145/2872427.2883062
34. Park, J.H., Fung, P.: One-step and two-step classification for abusive language detection on twitter. In: Proceedings of the First Workshop on Abusive Language Online, pp. 41–45. Association for Computational Linguistics (2017). URL http://aclweb.org/anthology/W17-3006
35. Pew Research Center: Online harassment (2017). URL http://www.pewinternet.org/2014/10/22/online-harassment/
36. Rahman, J.: The n word: Its history and use in the african american community. Journal of English Linguistics **40**(2), 137–171 (2012). DOI 10.1177/0075424211414807. URL https://doi.org/10.1177/0075424211414807
37. Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., Pande, V.: Massively multitask networks for drug discovery. arXiv preprint arXiv:1502.02072 (2015)
38. Roberts, D.E.: The social and moral cost of mass incarceration in african american communities. Stanford Law Review **56**(5), 1271–1306 (2004)
39. Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., Wojatzki, M.: Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. In: M. Beißwenger, M. Wojatzki, T. Zesch (eds.) Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication, *Bochumer Linguistische Arbeitsberichte*, vol. 17, pp. 6–9. Bochum (2016)
40. Smith, P.K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., Tippett, N.: Cyberbullying: its nature and impact in secondary school pupils. Journal of Child Psychology and Psychiatry **49**(4), 376–385 (2008). DOI 10.1111/j.1469-7610.2007.01846.x. URL http://dx.doi.org/10.1111/j.1469-7610.2007.01846.x
41. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642. Association for Computational Linguistics, Stroudsburg, PA (2013)

42. The Guardian: Germany approves plans to fine social media firms up to €50m (2017)
43. Waseem, Z.: Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In: Proceedings of the First Workshop on NLP and Computational Social Science, pp. 138–142. Association for Computational Linguistics, Austin, Texas (2016). URL http://aclweb.org/anthology/W16-5618
44. Waseem, Z., Davidson, T., Warmsley, D., Weber, I.: Understanding abuse: A typology of abusive language detection subtasks. In: Proceedings of the First Workshop on Abusive Language Online. Association for Computational Linguistics (2017)
45. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: Proceedings of the NAACL Student Research Workshop. Association for Computational Linguistics, San Diego, California (2016)
46. Wulczyn, E., Thain, N., Dixon, L.: Ex machina: Personal attacks seen at scale. In: Proceedings of the 26th International Conference on World Wide Web, WWW '17, pp. 1391–1399. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017). DOI 10.1145/3038912.3052591. URL https://doi.org/10.1145/3038912.3052591
47. Yu, J., Jiang, J.: Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. Association for Computational Linguistics (2016)