Joint Modelling of Emotion and Abusive Language Detection

Santhosh Rajamanickam
ILLC, University of Amsterdam
rajamanickamsanthosh@gmail.com

Pushkar Mishra
Facebook AI
pushkarmishra@fb.com

Helen Yannakoudakis
Dept.of Informatics, King's College London
helen.yannakoudakis@kcl.ac.uk

Ekaterina Shutova
ILLC, University of Amsterdam
e.shutova@uva.nl

Abstract

The rise of online communication platforms has been accompanied by some undesirable effects, such as the proliferation of aggressive and abusive behaviour online. Aiming to tackle this problem, the natural language processing (NLP) community has experimented with a range of techniques for abuse detection. While achieving substantial success, these methods have so far only focused on modelling the linguistic properties of the comments and the online communities of users, disregarding the emotional state of the users and how this might affect their language. The latter is, however, inextricably linked to abusive behaviour. In this paper, we present the first joint model of emotion and abusive language detection, experimenting in a multi-task learning framework that allows one task to inform the other. Our results demonstrate that incorporating affective features leads to significant improvements in abuse detection performance across datasets.

1 Introduction

Aggressive and abusive behaviour online can lead to severe psychological consequences for its victims (Munro, 2011). This stresses the need for automated techniques for abusive language detection, a problem that has recently gained a great deal of interest in the natural language processing community. The term *abuse* refers collectively to all forms of expression that vilify or offend an individual or a group, including *racism, sexism, personal attacks, harassment, cyber-bullying*, and many others. Much of the recent research has focused on detecting *explicit* abuse, that comes in the form of expletives, derogatory words or threats, with substantial success (Mishra et al., 2019b). However, abuse can also be expressed in more implicit and subtle ways, for instance, through the use of ambiguous terms and figurative language, which has proved more challenging to identify.

The NLP community has experimented with a range of techniques for abuse detection, such as recurrent and convolutional neural networks (Pavlopoulos et al., 2017; Park and Fung, 2017; Wang, 2018), character-based models (Nobata et al., 2016) and graph-based learning methods (Mishra et al., 2018a; Aglionby et al., 2019; Mishra et al., 2019a), obtaining promising results. However, all of the existing approaches have focused on modelling the linguistic properties of the comments or the meta-data about the users. On the other hand, abusive language and behaviour are also inextricably linked to the emotional and psychological state of the speaker (Patrick, 1901), which is reflected in the affective characteristics of their language (Mabry, 1974). In this paper, we propose to model these two phenomena jointly and present the first abusive language detection method that incorporates affective features via a multitask learning (MTL) paradigm.

MTL (Caruana, 1997) allows two or more tasks to be learned jointly, thus sharing information and features between the tasks. In this paper, our main focus is on abuse detection; hence we refer to it as the *primary task*, while the task that is used to provide additional knowledge — emotion detection — is referred to as the *auxiliary task*. We propose an MTL framework where a single model can be trained to perform emotion detection and identify abuse at the same time. We expect that affective features, which result from a joint learning setup through shared parameters, will encompass the emotional content of a comment that is likely to be predictive of potential abuse.

We propose and evaluate different MTL architectures. We first experiment with hard parameter sharing, where the same encoder is shared between the tasks. We then introduce two variants of the

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4270–4279

MTL model to relax the hard sharing constraint and further facilitate positive transfer. Our results demonstrate that the MTL models significantly outperform single-task learning (STL) in two different abuse detection datasets. This confirms our hypothesis of the importance of affective features for abuse detection. Furthermore, we compare the performance of MTL to a transfer learning baseline and demonstrate that MTL provides significant improvements over transfer learning.

2 Related Work

Techniques for abuse detection have gone through several stages of development, starting with extensive manual feature engineering and then turning to deep learning. Early approaches experimented with lexicon-based features (Gitari et al., 2015), bag-of-words (*BOW*) or *n-gram* features (Sood et al., 2012; Dinakar et al., 2011), and user-specific features, such as age (Dadvar et al., 2013) and gender (Waseem and Hovy, 2016).

With the advent of deep learning, the trend shifted, with abundant work focusing on neural architectures for abuse detection. In particular, the use of convolutional neural networks (*CNN*s) for detecting abuse has shown promising results (Park and Fung, 2017; Wang, 2018). This can be attributed to the fact that *CNN*s are well suited to extract local and position-invariant features (Yin et al., 2017). Character-level features have also been shown to be beneficial in tackling the issue of Out-of-Vocabulary (OOV) words (Mishra et al., 2018b), since abusive comments tend to contain obfuscated words. Recently, approaches to abuse detection have moved towards more complex models that utilize auxiliary knowledge in addition to the abuse-annotated data. For instance, Mishra et al. (2018a, 2019a) used community-based author information as features in their classifiers with promising results. Founta et al. (2019) used transfer learning to fine-tune features from the author metadata network to improve abuse detection.

MTL, introduced by Caruana (1997), has proven successful in many NLP problems, as illustrated in the MTL survey of Zhang and Yang (2017). It is interesting to note that many of these problems are domain-independent tasks, such as part-of-speech tagging, chunking, named entity recognition, etc. (Collobert and Weston, 2008). These tasks are not restricted to a particular dataset or domain, i.e., any text data can be annotated for the phenomena involved. On the contrary, tasks such as abuse detection are domain-specific and restricted to a handful of datasets (typically focusing on online communication), therefore presenting a different challenge to MTL.

Much research on emotion detection cast the problem in a categorical framework, identifying specific classes of emotions and using e.g., Ekman's model of six emotions (Ekman, 1992), namely anger, disgust, fear, happiness, sadness, surprise. Other approaches adopt the Valence-Arousal-Dominance (*VAD*) model of emotion (Mehrabian, 1996), which represents polarity, degree of excitement, and degree of control, each taking a value from a range. The community has experimented with a variety of computational techniques for emotion detection, including vector space modelling (Danisman and Alpkocak, 2008), machine learning classifiers (Perikos and Hatzilygeroudis, 2016) and deep learning methods (Zhang et al., 2018). In their work, Zhang et al. (2018) take an MTL approach to emotion detection. However, all the tasks they consider are emotion-related (annotated for either classification or emotion distribution prediction), and the results show improvements over single-task baselines. Akhtar et al. (2018) use a multitask ensemble architecture to learn emotion, sentiment, and intensity prediction jointly and show that these tasks benefit each other, leading to improvements in performance. To the best of our knowledge, there has not yet been an approach investigating emotion in the context of abuse detection.

3 Datasets

The tasks in an MTL framework should be related in order to obtain positive transfer. MTL models are sensitive to differences in the domain and distribution of data (Pan and Yang, 2009). This affects the stability of training, which may deteriorate performance in comparison to an STL model (Zhang and Yang, 2017). We experiment with abuse and emotion detection datasets[1] that are from the same data domain — Twitter. All of the datasets were subjected to the same pre-processing steps, namely lower-casing, mapping all *mentions* and *URLs* to a common token (i.e., *_MTN_* and *_URL_*) and mapping hashtags to words.

[1] We do not own any rights to the datasets (or the containing tweets). In the event of one who wishes to attain any of the datasets, to avoid redistribution infringement, we request them to contact the authors/owners of the source of the datasets.

3.1 Abuse detection task

To ensure that the results are generalizable, we experiment with two different abuse detection datasets.

OffensEval 2019 (*OffensEval*) This dataset is from *SemEval 2019 - Task 6: OffensEval 2019 - Identifying and Categorizing Offensive Language in Social Media* (Zampieri et al., 2019a,b). We focus on Subtask A, which involves offensive language identification. It contains $13,240$ annotated tweets, and each tweet is classified as to whether it is offensive (33%) or not (67%). Those classified as offensive contain offensive language or targeted offense, which includes insults, threats, profane language and swear words. The dataset was annotated using crowdsourcing, with gold labels assigned based on the agreement of three annotators.

Waseem and Hovy 2016 (*Waseem&Hovy*) This dataset was compiled by Waseem and Hovy (2016) by searching for commonly used slurs and expletives related to religious, sexual, gender and ethnic minorities. The tweets were then annotated with one of three classes: *racism*, *sexism* or *neither*. The annotations were subsequently checked through an expert review, which yielded an inter-annotator agreement of $\kappa = 0.84$. The dataset contains $16,907$ TweetIDs and their corresponding annotation, out of which only $16,202$ TweetIDs were retrieved due to users being reported or tweets having been taken down since it was first published in 2016. The distribution of classes is: $1,939$ (12%) *racism*; $3,148$ (19.4%) *sexism*; and $11,115$ (68.6%) *neither*, which is comparable to the original distribution: ($11.7\% : 20.0\% : 68.3\%$).

It should be noted that racial or cultural biases may arise from annotating data using crowdsourcing, as pointed out by Sap et al. (2019). The performance of the model depends on the data used for training, which in turn depends on the quality of the annotations and the experience level of the annotators. However, the aim of our work is to investigate the relationship between emotion and abuse detection, which is likely to be independent of the biases that may exist in the annotations.

3.2 Emotion detection task

Emotion (*SemEval18*) This dataset is from *SemEval-2018 Task 1: Affect in Tweets* (Mohammad et al., 2018), and specifically from Subtask 5 which is a multilabel classification of 11 emotion labels that best represent the mental state of the author of a tweet. The dataset consists of around 11k tweets (training set: 6839; development set: 887; test set: 3260). It contains the TweetID and 11 emotion labels (*anger*, *anticipation*, *disgust*, *fear*, *joy*, *love*, *optimism*, *pessimism*, *sadness*, *surprise*, *trust*) which take a binary value to indicate the presence or absence of the emotion. The annotations were obtained for each tweet from at least 7 annotators and aggregated based on their agreement.

4 Approach

In this section, we describe our baseline models and then proceed by describing our proposed models for jointly learning to detect emotion and abuse.

4.1 Single-Task Learning

As our baselines, we use different Single-Task Learning (STL) models that utilize abuse detection as the sole optimization objective. The STL experiments are conducted for each primary-task dataset separately. Each STL model takes as input a sequence of words $\{w_1, w_2, ..., w_n\}$, which are initialized with k-dimensional vectors e from a pre-trained embedding space. We experiment with two different architecture variants:

Max Pooling and MLP classifier We refer to this baseline as $\text{STL}_{maxpool+MLP}$. In this baseline, a two-layered bidirectional Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997) is applied to the embedding representations e of words in a post to get contextualized word representations $\{h_1, h_2, ..., h_n\}$:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (1)$$

with $\overrightarrow{h_t}, \overleftarrow{h_t} \in \mathbb{R}^l$ and $h_t \in \mathbb{R}^{2 \cdot l}$, where l is the hidden dimensionality of the BiLSTM. We then apply a max pooling operation over $\{h_1, h_2, ..., h_n\}$:

$$r_i^{(p)} = max_i(h_1, h_2, ..., h_n) \quad (2)$$

where $r^{(p)} \in \mathbb{R}^{2 \cdot l}$ and where the superscript (p) is used to indicate that the representations correspond to the primary task. This is followed by dropout (Srivastava et al., 2014) for regularization and a 2-layered Multi-layer Perceptron (MLP) (Hinton, 1987):

$$\mathrm{m}^{1(p)} = BatchNorm(tanh(W^{l_1}\mathrm{r}^{(p)})) \quad (3)$$

$$\mathrm{m}^{2(p)} = tanh(W^{l_2}\mathrm{m}^{1(p)}) \quad (4)$$

$$\mathrm{m}_t^{(p)} = \mathrm{m}_t^{2(p)} \quad (5)$$

where W^{l_1} and W^{l_2} are the weight matrices of the 2-layer MLP. Dropout is applied to the output $\mathrm{m}^{(p)}$ of the MLP, which is then followed by a linear output layer to get the unnormalized output $\mathrm{o}^{(p)}$. For *OffensEval*, a sigmoid activation σ is then applied in order to make a binary prediction with respect to whether a post is offensive or not, while the network parameters are optimized to minimize the binary cross-entropy (BCE):

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i)) \quad (6)$$

where N is the number of training examples, and y denotes the true and $p(y)$ the predicted label. For *Waseem&Hovy*, a $log\_softmax$ activation is applied for multiclass classification, while the network parameters are optimized to minimize the categorical cross-entropy, that is, the negative log-likelihood (NLL) of the true labels:

$$L_{NLL} = -\frac{1}{N}\sum_{i=1}^{N} log(p(y_i)) \quad (7)$$

BiLSTM and Attention classifier We refer to this model as $\text{STL}_{BiLSTM+attn}$. In this baseline (Figure 1; enclosed in the dotted boxes), rather than applying max pooling, we apply dropout to h which is then followed by a third BiLSTM layer and an attention mechanism:

$$\mathrm{u}_t^{(p)} = W^a \mathrm{r}_t^{(p)} \quad (8)$$

$$a_t^{(p)} = \frac{exp(\mathrm{u}_t^{(p)})}{\sum_t exp(\mathrm{u}_t^{(p)})} \quad (9)$$

$$\mathrm{m}^{(p)} = \sum_t a_t^{(p)} \mathrm{r}_t^{(p)} \quad (10)$$

where $\mathrm{r}^{(p)}$ is the output of the third BiLSTM. We then apply dropout to the output of the attention layer $\mathrm{m}^{(p)}$. The remaining components, output layer and activation, are the same as the $\text{STL}_{maxpool+MLP}$ model.

Across the two STL baselines, we further experiment with two different input representations: 1) GloVe (G), where the input is projected through the GloVe embedding layer (Pennington et al., 2014); 2) GloVe+ELMo (G+E), where the input is first projected through the GloVe embedding layer and the ELMo embedding layer (Peters et al., 2018) separately, and then the final word representation e is obtained by concatenating the output of these two layers. Given these input representations, we have a total of 4 different baseline models for abuse detection. We use grid search to tune the hyperparameters of the baselines on the development sets of the primary task (i.e., abuse detection).

4.2 Multi-task Learning

Our MTL approach uses two different optimization objectives: one for abuse detection and another for emotion detection. The two objectives are weighted by a hyperparameter β [$(1 - \beta)$ for abuse detection and β for emotion detection] that controls the importance we place on each task. We experiment with different STL architectures for the auxiliary task and propose MTL models that contain two network branches – one for the primary task and one for the auxiliary task – connected by a shared encoder which is updated by both tasks alternately.

Hard Sharing Model This model architecture, referred to as MTL_{Hard}, is inspired by Caruana (1997) and uses *hard parameter sharing*: it consists of a single encoder that is shared and updated by both tasks, followed by task-specific branches. Figure 1 presents MTL_{Hard} where the dotted box represents the $\text{STL}_{BiLSTM+attn}$ architecture that is specific to the abuse detection task. In the right-hand side branch – corresponding to the auxiliary objective of detecting emotion – we apply dropout to h before passing it to a third BiLSTM. This is then followed by an attention mechanism to obtain $\mathrm{m}^{(a)}$ and then dropout is applied to it. The superscript (a) is used to indicate that these representations correspond to the auxiliary task. Then, we obtain the unnormalized output $\mathrm{o}^{(a)}$ after passing $\mathrm{m}^{(a)}$ through a linear output layer with $\mathrm{o}^{(a)} \in \mathbb{R}^{11}$ (11 different emotions in *SemEval18*), which is then subjected to a sigmoid activation to obtain a prediction $p(y)$. While the primary task on the left is optimized using either Equation 6 or 7 (depending on the dataset used), the auxiliary task is optimized to minimize binary cross-entropy.

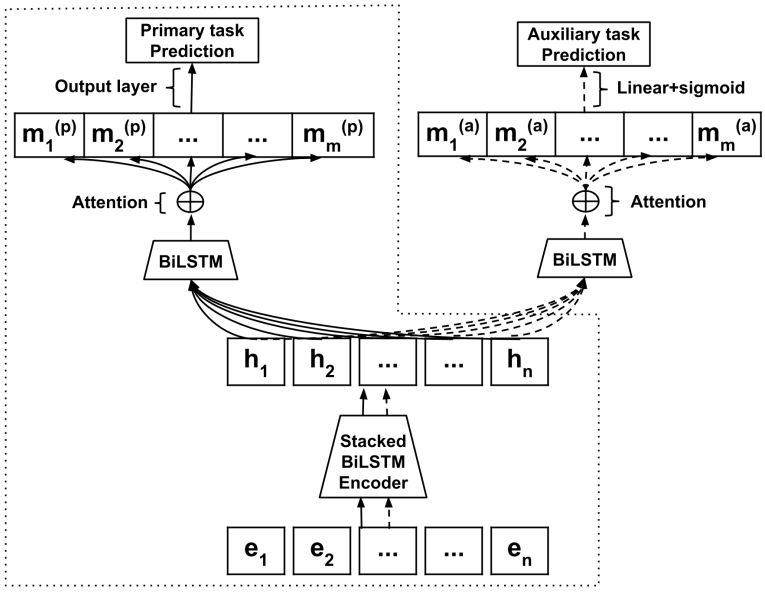

Figure 1: MTL *Hard Sharing* model. The embedding representations $\{e_1, e_2, ..., e_n\}$ are either a result of projection through the GloVe embedding layer or a concatenation of the projections through the GloVe and ELMo embedding layer. The different arrows are used to indicate the different passes for the primary and auxiliary task. The units on the left-hand side correspond to the primary task and the units on the right-hand side correspond to the auxiliary task with the *Stacked BiLSTM Encoder* and embedding layers shared by both tasks. The model inside the dotted box corresponds to the $\text{STL}_{BiLSTM+attn}$ architecture.

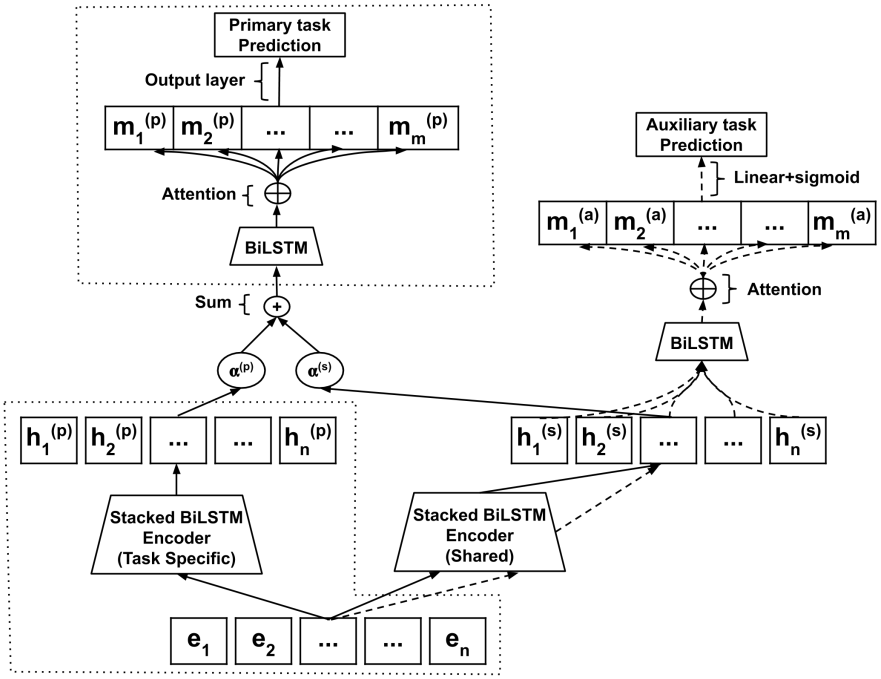

Figure 2: MTL *(Gated) Double Encoder* architecture. For the MTL Gated Double Encoder model we use two learnable parameters α that control information flow. For the MTL *Double Encoder* model, these are fixed and set to 1. The dotted boxes represent the $\text{STL}_{BiLSTM+attn}$ architecture.

Double Encoder Model This model architecture, referred to as $\text{MTL}_{DEncoder}$, is an extension of the previous model that now has two BiLSTM encoders: a task-specific two-layered BiLSTM encoder for the primary task, and a shared two-layered BiLSTM encoder. During each training step of the primary task, the input representation e for the primary task is passed through

both encoders, which results in two contextualized word representations $\{h_1^{(p)}, h_2^{(p)}, ..., h_n^{(p)}\}$ and $\{h_1^{(s)}, h_2^{(s)}, ..., h_n^{(s)}\}$, where superscript (s) is used to denote the representations that result from the shared encoder. These are then summed (Figure 2, where both $\alpha^{(p)}$ and $\alpha^{(s)}$ are fixed and set to 1) and the output representation is passed through a third BiLSTM followed by an attention mechanism to get the post representation $m^{(p)}$. The rest of the components of the primary task branch, as well as the auxiliary task branch are the same as those in MTL_{Hard}.

Gated Double Encoder Model This model architecture, referred to as $\text{MTL}_{GatedDEncoder}$, is an extension of $\text{MTL}_{DEncoder}$, but is different in the way we obtain the post representations $m^{(p)}$. Representations $h^{(p)}$ and $h^{(s)}$ are now merged using two learnable parameters $\alpha^{(p)}$ and $\alpha^{(s)}$ (where $\alpha^{(p)} + \alpha^{(s)} = 1.0$) to control the flow of information from the representations that result from the two encoders (Figure 2):

$$\alpha^{(p)} \cdot h^{(p)} + \alpha^{(s)} \cdot h^{(s)} \quad (11)$$

The remaining architecture components of the primary task and auxiliary task branch are the same as for $\text{MTL}_{DEncoder}$.

5 Experiments and results

5.1 Experimental setup

Hyperparameters We use pre-trained GloVe embeddings[2] with dimensionality 300 and pre-trained ELMo embeddings[3] with dimensionality 1024. Grid search is performed to determine the optimal hyperparameters. We find an optimal value of $\beta = 0.1$ that makes the updates for the auxiliary task 10 times less important. The encoders consist of 2 stacked BiLSTMs with $hidden\_size = 512$. For all primary task datasets, the BiLSTM+Attention classifier and the 2-layered MLP classifier have $hidden\_size = 256$. For the auxiliary task datasets, the BiLSTM+Attention classifier and the 2-layered MLP classifier have $hidden\_size = 512$. Dropout is set to 0.2. We use the *Adam optimizer* (Kingma and Ba, 2014) for all experiments. All model weights are initialized using *Xavier Initialization* (Glorot and Bengio, 2010). For $\text{MTL}_{GatedDEncoder}$, $\alpha^{(p)} = 0.9$ and $\alpha^{(s)} = 0.1$.

[2]https://nlp.stanford.edu/projects/glove/

[3]https://allennlp.org/elmo

STL model		P	R	F1
G	*maxpool+MLP*	76.35	**73.34**	74.24
	BiLSTM+attn	77.34	72.77	73.97
G+E	*maxpool + MLP*	77.19	72.73	73.95
	BiLSTM+attn	**77.40**	73.27	**74.40**

(a) *Twitter - OffensEval* STL results.

STL model		P	R	F1
G	***maxpool+MLP***	79.39	**78.20**	**78.33**
	BiLSTM+attn	77.97	77.57	77.49
G+E	*maxpool+MLP*	**80.66**	77.13	78.31
	BiLSTM+attn	79.08	77.93	78.16

(b) *Twitter - Waseem and Hovy* STL results.

Table 1: STL model comparisons. In these tables, G denotes models that use GloVe embeddings and G+E denotes models in which word representations are concatenations of their corresponding GloVe and ELMo embeddings. The best performing model is highlighted in bold.

Training All models are trained until convergence for both the primary and the auxiliary task, and early stopping is applied based on the performance on the validation set. For MTL, we ensure that both the primary and the auxiliary task have completed at least 5 epochs of training. The MTL training process involves randomly (with $p = 0.5$) alternating between the abuse detection and emotion detection training steps. Each task has its own loss function, and in each of the corresponding task's training step, the model is optimized accordingly. All experiments are run using stratified 10-fold cross-validation, and we use the paired t-test for significance testing. We evaluate the models using Precision (P), Recall (R), and F1 ($F1$), and report the average $macro$ scores across the 10 folds.

5.2 STL experiments

The STL experiments are conducted on the abuse detection datasets independently. As mentioned in the STL section, we experiment with four different model configurations to select the best STL baseline.

Table 1a presents the evaluation results of the STL models trained and tested on the *OffensEval* dataset, and Table 1b on the *Waseem and Hovy* dataset. The best results are highlighted in bold and are in line with the validation set results. We select the best performing STL model configuration on each dataset and use it as part of the corresponding MTL architecture in the MTL experiments below.

Model	P	R	F1
$\text{STL}_{BiLSTM+attn}$	77.40	73.27	74.40
MTL_{Hard}	77.21	73.30	74.51
$\text{MTL}_{DEncoder}$	**77.47**	73.82	74.97
$\text{MTL}_{GatedDEncoder}$	77.46	**75.27**	**76.03**†

(a) *Twitter - OffensEval* results.

Model	P	R	F1
$\text{STL}_{maxpool+MLP}$	79.39	78.20	78.33
MTL_{Hard}	79.34	77.61	77.90
$\text{MTL}_{DEncoder}$	**80.77**	78.18	79.02
$\text{MTL}_{GatedDEncoder}$	80.12	**79.60**	**79.55**†

(b) *Twitter - Waseem and Hovy* results.

Table 2: STL vs. MTL with emotion detection as the auxiliary task. † indicates statistically significant improvement over STL.

Dataset	Method	P	R	F1
OE	MTL	**77.46**	**75.27**†	**76.03**†
	Transfer	76.81	73.71	74.67
W&H	MTL	80.12	**79.60**†	**79.55**
	Transfer	**81.28**	77.72	79.07

Table 3: MTL vs. transfer learning performance. *OE* refers to the *OffensEval* dataset and *W&H* to the *Waseem&Hovy* dataset. † indicates statistically significant improvements.

5.3 MTL experiments

In this section, we examine the effectiveness of the MTL models for the abuse detection task and explore the impact of using emotion detection as an auxiliary task. We also compare the performance of our MTL models with that of a transfer learning approach.

Emotion detection as an auxiliary task In this experiment, we test whether incorporating emotion detection as an auxiliary task improves the performance of abuse detection. Tables 2a and 2b show the results on *OffensEval* and *Waseem and Hovy* datasets († indicates statistically significant results over the corresponding STL model). Learning emotion and abuse detection jointly proved beneficial, with MTL models achieving statistically significant improvement in F1 using the *Gated Double Encoder Model* $\text{MTL}_{GatedDEncoder}$ ($p < 0.05$, using a paired t-test). This suggests that affective features from the shared encoder benefit the abuse detection task.

MTL vs. transfer learning Transfer learning is an alternative to MTL that also allows us to transfer knowledge from one task to another. This experiment aims to compare the effectiveness of MTL against transfer learning. We selected the MTL model with the best performance in abuse detection and compared it against an identical model, but trained in a transfer learning setting. In this setup, we first train the model on the emotion detection task until convergence and then proceed by fine-tuning it for the abuse detection task. Table 3 presents the comparison between MTL and transfer learning, for which we use the same architecture and hyperparameter configuration as MTL. We observe that MTL outperforms transfer learning and provides statistically significant ($p < 0.05$) results on both *OffensEval* and *Waseem and Hovy* datasets.

6 Discussion

Auxiliary task Our results show that emotion detection significantly improves abuse detection on both *OffensEval* and *Waseem and Hovy* datasets. Table 4 presents examples of improvements in both datasets achieved by the $\text{MTL}_{GatedDEncoder}$ model, over the STL model. In the examples, the highlighted words are emotion evocative words, which are also found in the *SemEval2018 Emotion* dataset. As the emotion detection task encourages the model to learn to predict the emotion labels for the examples that contain these words, the word representations and encoder weights that are learned by the model encompass some affective knowledge. Ultimately, this allows the MTL model to determine the affective nature of the example, which may help it to classify abuse more accurately. It is also interesting to observe that a controversial person or topic may strongly influence the classification of the sample containing it. For example, sentences referring to certain politicians may be classified as *Offensive*, regardless of the context. An example instance of this can be found in Table 4.[4] The MTL model, however, classifies it correctly, which may be attributed to the excessive use of "*!*" marks. The latter is one of the most frequently used symbols in the *SemEval2018 Emotion* dataset, and it can encompass many emotions such as *surprise*, *fear*, etc., therefore, not being indicative of a particular type of emotion. Such knowledge can be learned within the shared features of the MTL model.

[4]We mask the name using the *_POLITICIAN_* tag.

Sample	STL	MTL	Gold Label	Predicted Emotion
Shut up *Katie and Nikki... That is all :) #HASHTAG*	*neither*	*sexism*	*sexism*	*disgust*
MTN That's the disadvantage of following a religion of uneducated ***morons****, so that you have to rely on Kufir for everything.*	*neither*	*racism*	*racism*	*anger*, *disgust*
MTN Earthly tyrants want to be ***feared*** *because for them* ***fear*** *is control and obedience. The writer of the Quran was unsophisticated.*	*neither*	*racism*	*racism*	*fear*, *optimism*
MTN And does this ***surprise*** *any of us _POLITICIAN_ SUPPORTERS!!! Not at all... We have heard him accused of everything that can be imagined!!! We still stand BEHIND _POLITICIAN_!!!*	*Offensive*	*NotOffensive*	*NotOffensive*	None
MTN I m pretty sure you are not too bad yourself... ***thanks*** *for a lil bit of* ***sweetness*** *on this brutal world*	*Offensive*	*NotOffensive*	*NotOffensive*	*joy*, *optimism*

Table 4: STL vs. MTL: samples from *Twitter - Waseem and Hovy* and *Twitter - OffensEval* datasets, where superior performance of MTL is observed. The 'predicted emotion' column contains the emotion labels predicted on the abuse detection data. The name of the politician in the fourth row is masked using the *_POLITICIAN_* tag.

MTL vs. transfer learning This experiment demonstrates that MTL achieves higher performance than transfer learning in a similar experimental setting. The higher performance may be indicative of a more stable way of transferring knowledge, which leads to better generalization. In the MTL framework, since the shared parameters are updated alternately, each task learns some knowledge that may be mutually beneficial to both related tasks, which leads to a shared representation that encompasses the knowledge of both tasks and hence is more generalized. In contrast, in the case of transfer learning, the primary task fine-tunes the knowledge from the auxiliary task (i.e., in the form of pre-trained parameters) for its task objective and may be forgetting auxiliary task knowledge.

7 Conclusion

In this paper, we proposed a new approach to abuse detection, which takes advantage of the affective features to gain auxiliary knowledge through an MTL framework. Our experiments demonstrate that MTL with emotion detection is beneficial for the abuse detection task in the *Twitter* domain. The mutually beneficial relationship that exists between these two tasks opens new research avenues for improvement of abuse detection systems in other domains as well, where emotion would equally play a role. Overall, our results also suggest the superiority of MTL over STL for abuse detection. With this new approach, one can build more complex models introducing new auxiliary tasks for abuse detection. For instance, we expect that abuse detection may also benefit from joint learning with complex semantic tasks, such as figurative language processing and inference.

References

Guy Aglionby, Chris Davis, Pushkar Mishra, Andrew Caines, Helen Yannakoudakis, Marek Rei, Ekaterina Shutova, and Paula Buttery. 2019. CAMsterdam at SemEval-2019 task 6: Neural and graph-based feature extraction for the identification of offensive tweets. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 556–563, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Md Shad Akhtar, Deepanway Ghosal, Asif Ekbal, Pushpak Bhattacharyya, and Sadao Kurohashi. 2018. A multi-task ensemble framework for emotion, sen-

timent and intensity prediction. *arXiv preprint arXiv:1808.01216*.

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

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *European Conference on Information Retrieval*, pages 693–696. Springer.

Taner Danisman and Adil Alpkocak. 2008. Feeler: Emotion classification of text using vector space model. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, page 53.

Karthik Dinakar, Roi Reichart, and Henry Lieberman. 2011. Modeling the detection of textual cyberbullying. In *fifth international AAAI conference on weblogs and social media*.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Antigoni Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2019. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM Conference on Web Science*, pages 105–114. ACM.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

Geoffrey E Hinton. 1987. Learning translation invariant recognition in a massively parallel networks. In *International Conference on Parallel Architectures and Languages Europe*, pages 1–13. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Edward A Mabry. 1974. Dimensions of profanity. *Psychological Reports*, 35(1):387–391.

Albert Mehrabian. 1996. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018a. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1088–1098, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019a. Abusive Language Detection with Graph Convolutional Networks. 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 2145–2150, Minneapolis, Minnesota. Association for Computational Linguistics.

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2018b. Neural character-based composition models for abuse detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 1–10, Brussels, Belgium. Association for Computational Linguistics.

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

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17.

Emily R. Munro. 2011. The protection of children online: a brief scoping review to identify vulnerable groups.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *WWW*, pages 145–153.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*.

George TW Patrick. 1901. The psychology of profanity. *Psychological Review*, 8(2):113.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deep learning for user comment moderation. In *ALW*, pages 25–35.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Isidoros Perikos and Ioannis Hatzilygeroudis. 2016. Recognizing emotions in text using ensemble of classifiers. *Engineering Applications of Artificial Intelligence*, 51:191–201.

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*.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678.

Sara Owsley Sood, Judd Antin, and Elizabeth Churchill. 2012. Using crowdsourcing to improve profanity detection. In *2012 AAAI Spring Symposium Series*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Cindy Wang. 2018. Interpreting neural network hate speech classifiers. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 86–92.

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

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.

Yuxiang Zhang, Jiamei Fu, Dongyu She, Ying Zhang, Senzhang Wang, and Jufeng Yang. 2018. Text emotion distribution learning via multi-task convolutional neural network. In *IJCAI*, pages 4595–4601.