Journal of Intelligent Information Systems (2023) 60:495–519
https://doi.org/10.1007/s10844-022-00726-4

Multi-task learning for toxic comment classification and rationale extraction

Kiran Babu Nelatoori[1] · **Hima Bindu Kommanti**[1]

Received: 21 April 2022 / Revised: 29 June 2022 / Accepted: 30 June 2022 /
Published online: 20 August 2022

Abstract
Social media content moderation is the standard practice as on today to promote healthy discussion forums. Toxic span prediction is helpful for explaining the toxic comment classification labels, thus is an important step towards building automated moderation systems. The relation between toxic comment classification and toxic span prediction makes joint learning objective meaningful. We propose a multi-task learning model using ToxicXLMR for bidirectional contextual embeddings of input text for toxic comment classification, and a Bi-LSTM CRF layer for toxic span or rationale identification. To enable multi-task learning in this domain, we have curated a dataset from Jigsaw and Toxic span prediction datasets. The proposed model outperformed the single task models on the curated and toxic span prediction datasets with 4% and 2% improvement for classification and rationale identification, respectively. We investigated the domain adaptation ability of the proposed MTL model on HASOC and OLID datasets that contain the out of domain text from Twitter and found a 3% improvement in the F1 score over single task models.

Keywords Multi-Task Learning (MTL) · Joint loss · Toxic Span Prediction (TSP) · Toxic Comment Classification (TCC) · Rationale extraction · Transfer learning

1 Introduction

The freedom of speech and expression offered by social media platforms is misused by some people to fill these platforms with abusive content. Though adults can manage this menace to some extent, children, and teens are susceptible to serious mental health issues,

✉ Hima Bindu Kommanti
himabinduk@nitandhra.ac.in

Kiran Babu Nelatoori
kiranbabu.sclr@nitandhra.ac.in

[1] Department of CSE, National Institute of Technology Andhra Pradesh, 534101 Andhra Pradesh, India

as reported by Temper et al. (2013), Sonone et al. (2021), and Dellerman (2022). There has been a 70% increase in the amount of bullying/hate speech among teens and children since the Covid-19 lockdown.[1] This has resulted in rising interest in artificial intelligence and natural language processing community relating to social and ethical challenges, which has been fueled by the worldwide commitment to combat toxic content. This toxic content is synonymous with hate, offensive, abusive, cyberbullying, violence, and other online forms of harassment. The growing interest in addressing this menace of toxic content by the computer science community in recent times is evident from the workshops such as TRAC 2020,[2] STOC 2020[3] and WOAH6[4] to be held in 2022. Most of the social media platforms follow content moderation to restrict toxic content. Due to the massive scale of the online content, we need unbiased and scalable systems to detect toxic content in real-time. These systems will gain people's confidence if they identify the span of text that is responsible for classifying the content as toxic. Toxic free social media platforms are needed to promote healthy discussions among the people.

Earlier research in this domain focused on identifying whether the entire content is toxic or not by classification methods. These models range from machine learning models such as logistic regression, support vector machines (Waseem and Hovy, 2016; Davidson et al., 2017) and deep learning models such as CNN, RNN, and Attention networks (Park & Fung, 2017; Badjatiya et al., 2017; Founta et al., 2019; Chakrabarty et al., 2019; Chia et al., 2021; Kiran Babu & HimaBindu, 2022). Convolutional Recurrent Neural networks (Ashok Kumar et al., 2021; Elnaggar et al., 2018; Zhang et al., 2018) are also employed to capture long-term dependencies in social media text. To improve the classification performance, transformer models are employed in Mozafari et al. (2019), Caselli et al. (2021), and Fortuna et al. (2021). These machines' existing inability to explain their judgments and actions to human users limits the efficacy of these systems. Thus, there is a need to develop self-contained, and explainable systems. For toxic comment classification, the toxic span serves as the rationale. Recent research (Mathew et al., 2021; Xiang et al., 2021) focused on explainable toxic comment classification by predicting the span of the comment. According to Adadi and Berrada (2018), explanations can be used to justify the decision, and improve the accuracy and transparency of the model.

Toxic comment classification (TCC) and toxic span prediction (TSP) are related tasks. Multi-task models have shown better performance when related tasks are trained together (Gong et al., 2019; Ed-drissiya et al., 2021). Multi-task Learning (MTL) jointly learns from multiple related tasks. MTL can be viewed as a type of inductive transfer and can improve the model's generalization for individual tasks by exchanging representations (e.g., shared parameters) between similar tasks. According to Baxter (2000), inductive transfer can assist enhancement of a model by providing an inductive bias that causes the model to favor one hypothesis over others.

In this paper, we propose a multi-task neural network model for toxic comment classification and rationale extraction. This multi-task neural network model jointly learns on sequence classification and span prediction tasks, and improved the performance of both tasks. Section 6 shows that the accuracy and F1 scores are better for

[1] https://enough.org/stats_cyberbullying
[2] https://sites.google.com/view/trac2/home
[3] https://social-threats.github.io/
[4] https://www.workshopononlineabuse.com/

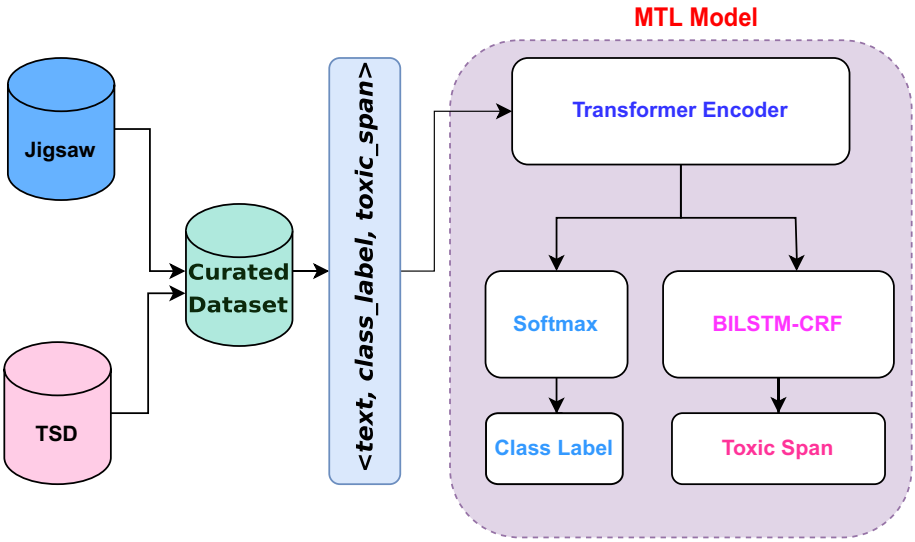

Fig. 1 Toxic comment classification and toxic span prediction system

both the classification and span prediction of the proposed MTL model. We curated a dataset from Jigsaw[5] and TSD (Toxic Span Detection) (Pavlopoulos et al., 2021) datasets as shown in Fig. 1 to enable multi-task learning. The Jigsaw dataset contains social media text with annotated class labels. The TSD dataset contains all toxic comments annotated with toxic spans. The model is trained on the curated dataset containing the triple $< text, class\_label, toxic\_span >$. The trained model's goal is to predict the class label (toxic or non-toxic) and toxic span when the social media text is given as input.

Our experimental results on the curated dataset and TSD dataset (Pavlopoulos et al., 2021) shows that our single MTL model improves the performance of both the classification and toxic spans prediction. In SemEval-2021 Task5 (Pavlopoulos et al., 2021), ensemble models are the winners. We claim that the proposed single MTL model is competitive enough with these ensemble models. In general, a model which is trained on a large coverage dataset in one domain and tested on a smaller coverage dataset in another domain will give better performance. When testing on unseen data, the proposed model is competitive enough with in-domain models. This shows that the proposed multi-task model not only improves the prediction performance but also improves the domain adaptation. We tested the transferability of the model trained on curated dataset on the unseen datasets viz. HASOC[6] and OLID.[7] This paper uses the words' *rationale*, *explanation* and *toxic span* interchangeably.

1.1 Research questions

This paper addresses the following research questions:

[5] https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/data
[6] https://hasocfire.github.io/hasoc/2019/dataset.html
[7] https://scholar.harvard.edu/malmasi/olid

RQ1 *What neural network models are beneficial for toxic comment classification and rationale extraction?* We examine Single Task Learning (STL) methodologies in deep learning and present an MTL architecture based on past research. We also use two STL baseline models to offer a more detailed examination.

RQ2 *What datasets are available to train and evaluate the MTL model?* No datasets are publicly available as per our knowledge that are compatible with MTL. We curated a dataset from two publicly available datasets.

RQ3 *Does MTL improve the performance of toxic comment classification and rationale extraction?* To answer this question, we compare the proposed MTL with STL baselines and assess the influence of MTL on the two related tasks by measuring their accuracy and F1 scores.

RQ4 *What pretrained language models are suitable for toxic comment classification and rationale extraction?* The word representations are crucial for every NLP task. Hence, we experimented with three publicly available pretrained transformer models and their fine-tuned variants.

RQ5 *Does MTL improve domain adaptation and generalizability of the model?* We collected two publicly available datasets which include hate or offensive texts to evaluate the proposed MTL and STL baselines to answer this question.

1.2 Contributions

This paper's contributions are as follows:

1. We present a multi-task learning model by leveraging the joint learning of toxic comment classification and toxic span prediction. The model uses a transformer-based Bi-LSTM CRF layer for these tasks to answer RQ1.
2. We curate a dataset of 29623 comments consisting of class labels and toxic spans to enable multi-task learning. This dataset is curated from the Jigsaw and Toxic span detection (TSD) datasets. (RQ2)
3. We provide an empirical analysis of the classification and span prediction performance of the multi-task model to answer RQ3.
4. We provide results and analysis of six transformer- based models, domain adaptation experiments on unseen datasets, and error analysis to understand the limitations of the proposed model, thereby answering RQ4 and RQ5.

The rest of the paper is organized as follows. We provide a brief summary of different explainable methods: post-hoc (gradient-based, perturbation based, attention-based) and constitutive methods where human rationales are used to learn explanations (Section 2). Section 3 briefly describes the usage of MTL. Section 4 provides the multi-task model used to jointly learn from toxic comment classification and toxic span prediction tasks. The experimental results are presented in Section 5, followed by result analysis (Section 6). Section 7 contains discussion of the results and the error analysis, and Section 8 concludes the paper with a summary and future works.

2 Related works

Machine learning explainability has lately received a lot of attention owing to the necessity for transparency. Existing explainable methodologies are classified into two types: post-hoc explainability and constitutive explainability. The goal of post-hoc explainability is

to provide explanations for existing models. In this category, LIME (Ribeiro et al., 2016) is a representational approach that uses an explainable model to approximate model decisions in the local area of the feature space. Gradient based techniques (Karen et al., 2014; Sundararajan et al., 2017) find relevant properties by calculating the gradient of an output concerning an input feature and estimate the contribution of various input features. Attention mechanisms (Vaswani et al., 2017; Xu et al., 2015) can also be used as explanations, which identify sections of the input that are attended by the model for specific predictions. Attention mechanisms are a more prevalent way of explaining individual predictions. This attention mechanism has played a key role in NLP, not just for explainability, but also for improving model performance (Devlin et al., 2019). However, Wiegreffe and Pinter (2019) recently called explainability effectiveness into question by pointing out that attention distributions are inconsistent with the significance of input units as measured by gradient-based methods.

In the second category, the goal of constitutive explainability is to build self-explanatory models. This is accomplished by including explainability restrictions through loss function into the model's learning process by the use of human rationales (Xiang et al., 2021; Zaidan et al., 2007). Zaidan et al. (2007) first proposed the use of rationales, in which human annotators highlight a section of text that justifies their labelling judgement. These extended reasoning annotations are used on a lesser amount of training data by these authors to improve sentiment categorization.

The majority of the existing works on toxicity detection have focused on enhancing model performance, with little emphasis on explainability. For explainable toxicity classification, Mathew et al. (2021) provided a benchmark annotated dataset with rationales and built an explainable hate speech detection model. Similarly, Pavlopoulos et al. (2021) provide an annotated dataset containing toxic comments with span labels. Due to the nature of the task given, most of the submissions (Bansal et al., 2021; Sharma et al., 2021; Zou and Li, 2021; Khan et al., 2021; Nguyen et al., 2021; Zhu et al., 2021) for SemEval-2021 Task 5 are aimed at identifying the toxic spans and does not provide the entire classification label.

MTL methodologies, recently utilized for sentiment analysis (Akhtar et al., 2020; Wang et al., 2021) and adverse drug events extraction (Ed-drissiya et al., 2021) have shown state-of-the-art performance. Xiang et al. (2021) first utilized the MTL approach for detecting explainable toxicity in social media posts. This model uses BERT and jointly learns from sequence labels and span labels. The authors considered predicting span as a token classification problem and tokens are identified as part of the span or not based on toxicity score. They have used the Mean Squared Error (MSE) loss to train the model. This model makes a local decision at every point of the sequence. Each token classification is independent of other tokens' classification. Moreover, the class label of the entire sequence is identified by the toxic span scores. If there is indirect toxicity where toxic scores of tokens are low, this model is unable to predict the toxic classes even though the text is toxic. As tokens of the toxic spans depend on each other, instead of identifying individual tokens as part of the toxic span, we modelled this problem as a sequence tagging problem, inspired from (Chen et al., 2019; Ed-drissiya et al., 2021). Conditional Random Fields (CRF) are used to predict token labels by taking the full sequence into account. This is especially beneficial in NLP applications where word sequences depend on each other and certain word sequences are implausible. Hence, we devised a transformer-based Bi-LSTM CRF network to predict the toxic span and the class label of the social media post. The toxic span is the cause for labelling the post as toxic, and it is empty string for non-toxic posts.

The introduction of transfer learning has undoubtedly aided in the acceleration of NLP research. We can leverage a pretrained model generated on a massive dataset and adjust it for different tasks on a task-specific dataset. Transformer-based pretrained language models have proven to be effective for various NLP tasks. A prime example is BERT (Devlin et al., 2019), which employs bidirectional transformer architecture to learn word association during pre-training, utilizing Masked Language Modelling and Next Sentence Prediction tasks for self-supervised learning. With an embedding size of 768, the transformer-based embeddings provide semantic and syntactic information of the input tokens. RoBERTa (Liu et al., 2019b) is a hyperparameter fine-tuned variant of BERT, and it is pretrained on a 160 GB corpus using a dynamic masking method. RoBERTa doesn't use Next Sentence Prediction for learning. To work with multilingual data, XLM-RoBERTa (XLMR) (Conneau et al., 2020) was trained on more than 2 TB of Common Crawl data of 100 different languages. Transformer-based models are recently being used for fake news detection and censored tweets classification (Mehta et al., 2021; Ahmed and Kumar M., 2021) for better prediction performance. In this paper, BERT, RoBERTa and XLMR are used in all the experiments to learn contextual embeddings of the input sequence. We have also experimented with fine-tuned versions of these models viz., ToxicBERT,[8] ToxicRoBERTa[9] and ToxicXLMR[10] to test the effect of fine-tuned transformers on the proposed task.

3 Multi-task learning

Multi-task learning (MTL) utilizes multiple similar tasks that can regularize each other to improve the performance of the target task. MTL has its roots in Caruana's pioneering research (Caruana 1993, 1997). MTL is subsequently used in a wide variety of machine learning applications, including Computer Vision (Long et al., 2017), Bioinformatics (Ramsundar et al., 2015), and various subfields of natural language processing (Worsham & Kalita, 2020; McCann et al., 2018). The basic idea behind MTL is to train a model that provides outputs for multiple related tasks based on a single common input. We contrast this with traditional machine learning techniques, in which a model is often a function from a single input space to a single output space. The reasoning behind the MTL concept is that information captured in training data for one task may assist the model to generalize better when learning on another related task.

From a theoretical viewpoint, MTL has the advantage of acting as a regularizer for a specific task to build generalized models that can handle unseen data. More precisely, as we optimize parameters for many tasks at the same time, the additional details contained in the auxiliary tasks function as a means to prevent the model from overfitting to the training data. Another benefit of MTL that we have used in our study is its capacity to learn from several related datasets. This implies that we may integrate datasets from a variety of jobs without having to re-annotate the data (to ensure that the label spaces are consistent). MTL is mostly preferred when the target tasks improve performance when compared to the single-task model (Standley et al., 2020).

[8] https://huggingface.co/unitary/toxic-bert
[9] https://huggingface.co/unitary/unbiased-toxic-roberta
[10] https://huggingface.co/unitary/multilingual-toxic-xlm-roberta

When selecting a set of tasks for MTL, some design considerations for modelling and training data are influenced by the relative priority of the tasks. If we are just interested in a single job; we can simply optimize our model to produce the greatest possible performance for that particular activity. When we are equally interested in good performance across all tasks, our work becomes significantly more difficult, since we require to strike a compromise between performance ratings across all the activities. It isn't always advantageous in enhancing a classifier's performance (Worsham & Kalita, 2020). Aside from the increasing complexity of the model and training time, the relationship between tasks and datasets is crucial for MTL effectiveness. After deciding on the model's design, we must determine the loss function for optimization. The most basic technique is to minimize a linear combination of the loss functions of each task. Every task has its own loss function (L_{task}). In our multi-task approach, we simply weigh each loss function and minimize the custom loss function as shown in (1), where w_i is the weight of i_{th} task's loss L_i. The simplest method is uniform weighing (Gong et al., 2019).

$$\min_{\theta} \sum_i w_i L_i(\theta) \tag{1}$$

The construction of a training scheme is the last issue of MTL training implications. The conventional MTL model training process is to create mini-batches containing samples for a single task and then switch between tasks throughout training. The percentage of task mini-batches might be the same for all tasks, or it can vary depending on task performance or dataset size (Caruana, 1997; McCann et al., 2018; Ruder, 2017). During the training phase, we may switch between optimization processes (Xiang et al., 2021) by having a suitable loss function for each task. Another issue that must be considered is how a task is expressed in the model. Each job in MTL has unique task-specific output layers for the task-specific outputs (Ruder, 2017). Instead of having multiple models per task and aggregating their results, the MTL model uses the shared parameter concept and reduces the complexity of the model. We only require task-specific heads for each task instead of having a new model per task, thereby reducing the overall complexity of the model.

4 Model description

We adopt an MTL model that can jointly learn and transform knowledge between toxic comment classification and toxic span prediction tasks. While a variety of techniques for MTL have been investigated in the past, the paradigm that has received the most interest in deep learning and natural language processing (NLP) in particular is *hard parameter sharing*. This MTL paradigm operates by sharing a fraction of the model's parameters between multiple tasks. We will compare the performance of these MTL models to that of STL baseline models that do not share any parameters.

Both the toxicity classification and toxic span prediction are related tasks. To reap benefits from these inter-related tasks, similar to Xiang et al. (2021) and Chen et al. (2019), we have built a multi-task learning neural network model to jointly train the model for both sequence classification and toxic span prediction tasks. MTL model for explaining toxicity label prediction shown is in Fig. 2.

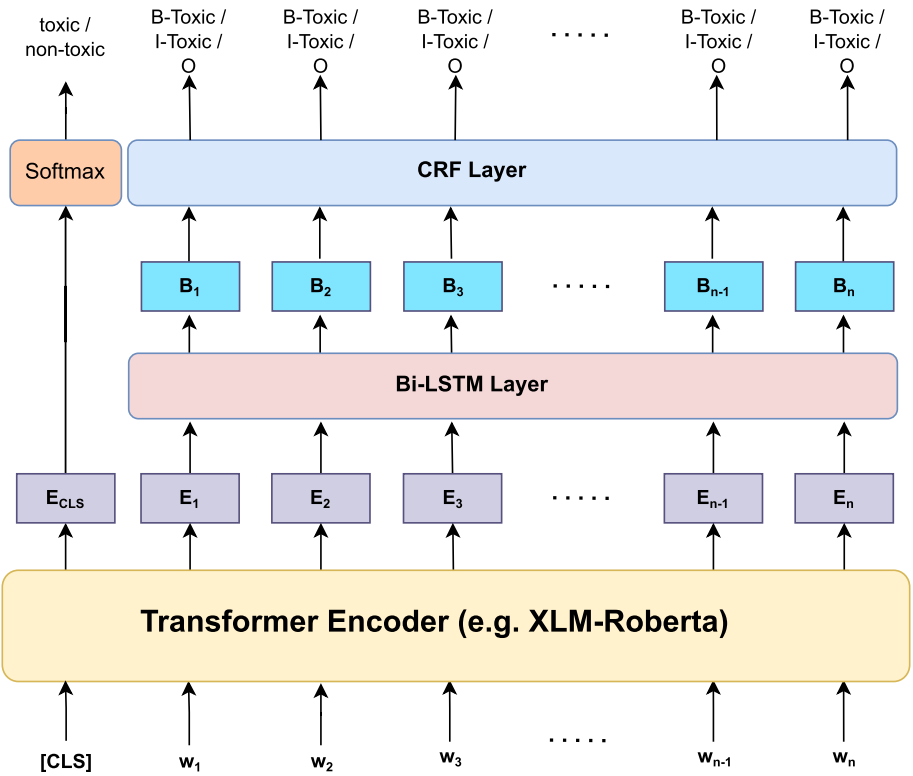

Fig. 2 Multi Task learning model for Toxicity Classification and Rationale Extraction

4.1 Problem statement

The social media post is designated by $I = \{w_1, w_2, \ldots, w_n\}$, where n is the sentence length. For any input sequence I, the task is to identify the class label $c_i \in C$, where $C = \{non\text{-}toxic, toxic\}$ and for each word $w_i \in I$ assign a tag $y_i^s \in Y^s$, where $Y^s = \{B\text{-}T, I\text{-}T, O\}$ to predict toxic span (rationale) of input sequence. To predict toxic spans, the *BIO* tagging scheme is used, where *B-T* (Begin) represents the first token in a toxic span, *I-T* (Inside) represents the inside and end tokens in a toxic span, and *O* represents the no-toxic tokens.

4.2 Proposed Model

A Transformer encoder is utilized to obtain the contextual embeddings of the input. To work with the transformer (e.g., BERT, XLM-Roberta), each input sequence is tokenized using the WordPiece tokenizer (Schuster and Nakajima, 2012). The special tokens viz., [*CLS*] and [*SEP*] are added at the beginning and end of the input sequence, respectively. [*CLS*] is used as a classification token and [*SEP*] is used to represent the end of the input sequence. Let C_t and Y_t be the number of distinct class and labels, respectively. Given a tokenized input sequence $I = \{[CLS], w_1, w_2, \ldots, w_n, [SEP]\}$, the output of the transformer encoder module will be $O = \{E_{CLS}, E_2, E_3, \ldots, E_N\}$ of size $[N, D_o] \in \mathbb{R}^{N \times D_o}$ where $N = n + 2$ and $D_o = 768$. D_o is the final hidden layer dimension of the transformer encoder.

$$O = TransformerEncoder(I) \tag{2}$$

The output of the transformer encoder contains contextual embeddings of each input token w_i. The special token *CLS* embedding $E_{CLS} \in \mathbb{R}^{D_o}$ is used for the classification task, as it contains the contextualized information of the entire input sentence, I. To avoid overfitting, we added a dropout layer with dropout rate 0.1 on the output of the transformer encoder. As shown in (3), this E_{CLS} embedding is fed to a linear layer with Softmax activation to predict class label y^c of the input sequence I.

$$y^c = Softmax\left(W_c^T.E_{CLS} + b_c\right) \tag{3}$$

where $W_c \in \mathbb{R}^{C_t \times D_o}$ is the weight matrix and $b_c \in \mathbb{R}^{D_o}$ is the bias vector.

BiLSTM-CRF models have shown promising results for NER and sequence tagging tasks (Huang et al., 2015; Ma & Hovy, 2016; Zou & Li, 2021). The transformer encoder works in parallel on the entire input sequence. Hence, we use a stacked Bi-LSTM layer on the output of the transformer encoder to obtain position-sensitive embeddings as shown in (4). This is necessary to use the sequence of the input tokens in determining the toxic span.

$$B = BiLSTM(O) \tag{4}$$

Let $B = \{B_1, B_2, ..., B_N\} \in \mathbb{R}^{N-1 \times D_b}$ denote the output features retrieved by the BiLSTM layer, where D_b is the hidden dimension of the BiLSTM layer.

$$S_i = LinearLayer(W_{s_i}^T.B_i),\ W_{s_i} \in \mathbb{R}^{Y_t \times D_b}, i \in \{2, \ldots N-1\} \tag{5}$$

A linear layer is added on top of the BiLSTM layer to get the label score $S_i \in \mathbb{R}^{Y_t}$ of each token w_i. W_{s_i} is the weight matrix and $S = \{S_1, S_2, \ldots S_{N-1}\}$ is the set of label scores of the input sequence. Toxic span prediction is influenced by the surrounding word predictions. The purpose of the CRF layer is to decode the best label chain $y^s = \{y_1^s, y_2^s, ..., y_N^s\}$ using S. CRF benefits from considering the correlations between labels/tags in the neighborhood as a discriminant graphical model, which is extensively utilized in sequence labelling or tagging applications (Ma & Hovy, 2016). The probabilistic CRF model defines a family of conditional probabilities $p(y|S)$ on all possible label sequences y given S.

We employ maximum conditional likelihood estimation for CRF training. The log-likelihood function is given by (6) and maximum likelihood training updates the parameters to maximize the log-likelihood L.

$$L = \sum_i log\ p(y|S) \tag{6}$$

The goal of decoding is to find the label sequence y^s with the highest conditional probability

$$y^s = \arg\max_{y \in Y(B)} p(y|S) \tag{7}$$

where $Y(B)$ represents all possible output sequences for input I. The CRF layer uses the Viterbi algorithm (Viterbi, 2009) to decode the label sequence by determining the most likely sequence of hidden states with the best posterior probability estimates.

4.3 Loss and model training

To jointly learn from both classification and span prediction tasks, the loss function is given by:

$$Loss = \alpha \sum L_{TC} + (1 - \alpha) \sum L_{TSP} \tag{8}$$

The weight parameter α controls the importance of each task.

The maximization of (6) is converted to a minimization problem by taking a negative log-likelihood ($-L$). The model is trained end-to-end, by minimizing a weighted loss of both tasks. Cross entropy loss (L_{TC}) and CRF loss ($L_{TSP} = -L$) are used for classification and toxic span prediction tasks, respectively. The overall training process of the model is summarized in *Algorithm 1*.

Algorithm 1 Multi-task model Training Algorithm

Input: A set of input sequences D_1 and D_2 from social media domain, $I = \{w_1, w_2, \ldots, w_N\}$ is one of the sample from D. Set D_1 contains span and class label information, and set D_2 contains only class label.

Output: Learned multi-task model

Initialize all learnable parameters Θ
$flag \leftarrow 0$
repeat
  for $b = 1 : n\_batches$ **do**
    if $flag = 0$ **then**
      Generate a batch of samples S_b from $D1$
    else
      Generate a batch of samples S_b from $D2$
    end if
    for each input sequence $I \in S_b$ **do**
      $O \leftarrow TransformerEncoder(I)$
      $y^c \leftarrow Softmax\,(W_c.E_{CLS} + b_c)$ ▷ Predict class label y^c
      $B \leftarrow BiLSTM(O)$
      $S \leftarrow LinearLayer(B)$ ▷ Calculate distinct lable scores of each token
      $y^s \leftarrow \arg\max_{y \in Y(B)} p(y|S)$ ▷ Predict toxic span sequence y^s using CRF
    end for
    if $S_b \in D_1$ **then**
      $loss \leftarrow \alpha \sum L_{TC} + (1 - \alpha) \sum L_{TSP}$ ▷ Calculate joint loss
      $flag \leftarrow 1$
    else
      $loss \leftarrow L_{TC}$ ▷ Calculate cross entropy loss of classification task
      $flag \leftarrow 0$
    end if
    Use the back-propagation algorithm to update parameters Θ by minimizing the loss with the batch update mode
  end for
until stopping criteria is met

5 Experiments

We evaluated our models on the curated dataset for classification performance and on the TSD dataset for toxic span prediction performance. To test for domain adaptation, we evaluated our models on the HASOC and OLID English datasets.

5.1 Datasets

We collected four publicly available datasets, two of these are used to curate new dataset to enable MTL, and the remaining two are used to test domain adaptation and generalizability of the model on unseen dataset. Each dataset is described in the following sub-sections.

5.1.1 Kaggle-Jigsaw[11]

Civil Comments platform used crowd-sourced moderation and advanced community management tools to bring real-world social cues to comment sections. Civil Comments is the first commenting platform created with the goal of improving how people interact online. It was shut down at the end of 2017.[12] They have provided nearly 2 million public comments in an open archive so that researchers could better understand and promote civil discourse in online interactions. Jigsaw funded the project and had human raters annotate the data for several toxic conversational characteristics: *toxic, severe toxic, obscene, threat, insult, identity hate*. We converted this dataset into a binary classification dataset for our training and testing purposes by merging all forms of toxicity into a single class to suit our problem statement.

5.1.2 TSD

The toxic span detection dataset (Pavlopoulos et al., 2021) is a subset of the Jigsaw dataset containing 10k toxic comment samples labelled with toxic spans. Each sample of this dataset is annotated by three individuals. A span is labelled as toxic only when at least two annotators label it as toxic. The annotation process is done by the SemEval-2021 and released the dataset for Task5.

5.1.3 Curated dataset

The Kaggle-Jigsaw dataset contains only the class labels for the whole text sequence, and it contains both toxic and non-toxic sequences. The TSD dataset contains only toxic posts with their toxic span information, hence do not contain the class label as shown in Fig. 3. As MTL requires both, we curated a dataset that contains class labels and span information from Jigsaw and TSD datasets. We collected the top 20 words which are part of toxic spans in the TSD dataset, and collected non-toxic posts containing these words from the Jigsaw dataset. The inclusion of non-toxic posts with toxic words will make the model learn the non-toxic usage of toxic words (Xiang et al., 2021). When

[11] https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/data
[12] https://medium.com/@aja_15265/saying-goodbye-to-civil-comments-41859d3a2b1d

Kaggle-Jigsaw Samples	**Label** Toxic Non-toxic		**Text** *You are a fucking cunt, lady.* *Can't argue with that math!*
TSD Samples		**Toxic Span** [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]	**Text** *She's a bald faced liar and an embarrassment to the Republican party and Alaska*
Curated Samples	**Label** Toxic Toxic Non-toxic	**Toxic Span** [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22] Unknown [ ]	**Text** *She's a bald faced liar and an embarrassment to the Republican party and Alaska* *You are a fucking cunt, lady.* *Can't argue with that math!*

Fig. 3 Samples from Jigsaw contains toxic and non-toxic posts and do not contain toxic span ground truth. Samples from TSD dataset are all toxic and contain toxic span ground truth. The curated dataset is a mixture of TSD and Jigsaw datasets. The toxic span ground truth is empty set for non-toxic samples. It is *unknown* for the toxic samples of Jigsaw dataset. The toxic span is shown in red color. **Content disclaimer**: This figure and some of the subsequent pages contain toxic content which may be disturbing to some of the readers, the toxic content is used only for illustration purposes

Table 1 Curated dataset distribution: 10K samples collected from TSD, 19k samples collected from Jigsaw and split into train, test and valid sets

	From TSD (Toxic)	From Jigsaw (Non-Toxic)	From Jigsaw (Toxic)	Total
Train	7939	12000	3000	22939
Test	1994	3000		4994
dev/trial	690	1000		1690
				29623

collecting Jigsaw samples, we removed the samples that overlapped with the TSD dataset. The TSD dataset contains nearly 10k samples. To have balanced non-toxic posts, a total of 19k posts were collected from the Jigsaw dataset out of 1.8 million posts, in which 16k were non-toxic and 3k were toxic. While curating the dataset from the Jigsaw, samples having *toxicscore* ≤ 0.1 are considered as non-toxic and *toxicscore* ≥ 0.8 are considered as toxic posts. 10k posts were collected from the TSD dataset and all the posts are treated as toxic.

All posts in the TSD dataset have toxic span information. For all non-toxic posts from the Jigsaw, the span information is empty, which means there is no toxic span to learn from non-toxic posts. Therefore, these posts are not included in the test and trial set, as they are not useful for evaluating the span prediction performance. To have balanced toxic and non-toxic posts in the training set, we followed the train-test-valid splits as shown in Table 1. Hence, the training set has nearly 11k toxic posts and 12k non-toxic posts.

5.1.4 HASOC and OLID

The HASOC and OLID datasets comprise tweets that have hierarchical annotations. The first level annotations contain the labels *offensive* and *not offensive*. The next level annotations are for the type of offensive. In our evaluation, we used the first level annotations, which indicate

Table 2 Statistics of three datasets (Curated, HASOC, OLID)

	Text length (in chars)	Word count per sequence
Curated	298 ± 272	58 ± 52
HASOC	160 ± 78	30 ± 14
OLID	146 ± 78	24 ± 13

Text length and word count with mean ± standard deviation are mentioned here

whether the tweet is offensive or not. We use the test samples from OLID and, HASOC to evaluate model performance for out of domain data. From Table 2, it is observed that social media post (text) length and number of words per post of HASOC and OLID datasets are small compared to the curated set.

5.2 Pre-processing

The pre-processing stage eliminates URLs and combines repeated strings into a single character (e.g., "!!!!!!" is changed to "!"). We remove white spaces in toxic offsets (e.g., in the text "you are astupid", the leading white-space is also marked as part of the toxic span) and, singleton offsets (e.g., in the text "ab usive speech," only 'b' is marked as a toxic offset). These are removed as inconsistencies of the annotator. Following these steps, the text is tokenized with a custom tokenizer to preserve terms like "a$$" rather than tokenizing it as 'a', '$', '$'. After tokenization, *B-T* (begin toxic) is allotted to token if it is at the beginning of the toxic span and an *I-T* (inside toxic) allotted if it is within the toxic span; otherwise, the *O* is assigned (other). *O* is also assigned for all tokens in non-toxic comments. These tokens serve as ground-truth tokens for the CRF layer. To get a fixed length input sequence, we use post-padding for shorter sequences and truncation for longer sequences.

5.3 Model implementation

For comparison, we have built baselines that model a classification task alone and span prediction task alone.

5.3.1 Classification task

For these models, we use token embedding [*CLS*] for sequence classification. As shown in Fig. 2, $E_{[CLS]}$ embedding is fed to the Softmax layer to predict the class label of the input sequence. All transformer-based sequence classification models are trained with a cross-entropy (L_{TC}) loss function.

5.3.2 Span prediction task

For these models, we use all the token embeddings for span prediction. As shown in Fig. 2, all token embeddings (E_i) are fed to the CRF layer. All transformer-based span

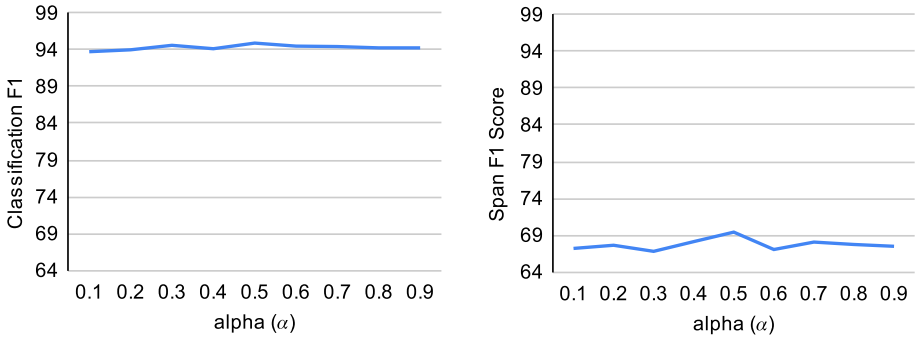

Fig. 4 Classification and Span Prediction F1 score vs. weight factor (α) of the loss function for Toxic XLMR – multi-task model

prediction models are trained to minimize span prediction loss (L_{TSP}). These models are trained on samples that contain only toxic span information.

5.3.3 Multi-task

The multi-task models are constructed as shown in Fig. 2 they are optimized using the loss function L indicated in (8). We use the joint loss (L) to compute the loss for the samples containing both post- and span-level labels. We compute only the classification loss L_{TC} for samples that contain only class labels for the entire text. During training, we created interleaved batches of both types of samples to ensure a balance update on the parameters.

5.3.4 BERT-MT

The BERT-MT model as proposed by Xiang et al. (2021) is replicated according to their description of model and hyperparameters. In this approach, input sequence is tokenized and fed to the BERT layer. On top of the BERT, a linear layer with sigmoid activation function is used for each token to find the toxic score of the token. The entire sequence toxic score is the maximum toxic score of its tokens. These scores are used to identify the class label and toxic span of the input sequence. The model is trained end-to-end using joint MSE loss.

5.4 Hyperparameters

We experimented with base case versions of BERT, RoBERTa, XLM-RoBERTa and their fine-tuned variants as embedding models. Hyperparameters were tuned using the development set. For all transformer models, the input sequence length is 512 tokens. Adam optimizer is used with a learning rate of $5e^{-05}$. The dropout rate is set to 0.1. The batch size is set to 24 and each model is trained end-to-end for 5 epochs. In the loss function (8), α is set to 0.5 to give equal importance to both tasks. Figure 4 shows the influence of α on the two

Table 3 Model Evaluation Results on Curated dataset

Transformer / Model	Results on Curated Test dataset		
	Accuracy	Weighted F1 Score	Macro F1 Score
Multi-task models (MTL)			
XLMR	93.12 ± 0.33	93.15 ± 0.32	92.90 ± 0.33
Toxic XLMR	**94.62 ± 0.27**	**94.64 ± 0.27**	**94.43 ± 0.28**
RoBERTa	93.40 ± 0.73	93.43 ± 0.72	93.20 ± 0.72
Toxic RoBERTa	93.81 ± 0.27	93.83 ± 0.26	93.59 ± 0.26
BERT	90.30 ± 0.43	90.32 ± 0.40	89.95 ± 0.38
Toxic BERT	90.84 ± 0.01	90.89 ± 0.01	90.56 ± 0.02
‡BERT-MT (Xiang et al., 2021)	90.63 ± 0.35	90.64 ± 0.33	90.26 ± 0.33
Classification task models (STL-C)			
XLMR	90.77 ± 0.38	90.76 ± 0.38	90.36 ± 0.39
ToxicXLMR	**90.99 ± 0.01**	**90.98 ± 0.02**	**90.56 ± 0.05**
RoBERTa	88.99 ± 0.60	89.01 ± 0.56	88.58 ± 0.54
ToxicRoBERTa	89.37 ± 0.10	89.35 ± 0.06	88.9 ± 0.020
BERT	90.52 ± 0.54	90.54 ± 0.52	90.15 ± 0.51
ToxicBERT	89.68 ± 0.46	89.69 ± 0.45	89.27 ± 0.50

‡ Replicated model

We report accuracy, weighted and macro F1 scores. The best performance is in bold. Each model is run three times with different seed values and their average is reported here

tasks. The best performance is observed at $\alpha = 0.5$. The influence of α is low on the classification task as the class labels can be predicted from the span predictions.

6 Result analysis

We ran a series of experiments to see how the three commonly used pretrained language models and their fine-tuned versions affected the overall performance of the proposed MTL system. This section provides the results of the proposed MTL and the baseline models for the toxic comment classification and toxic span prediction tasks.

6.1 Classification performance

The experimental results of MTL and STL for classification (STL-C) are shown in Table 3. Toxic versions of BERT, RoBERTa and XLMR are the fine-tuned versions of BERT, RoBERTa and XLMR respectively that were trained for the classification task. These are available at Hugging Face.[13] ToxicXLMR based multi-task model achieved the best performance compared to all the models we have evaluated. From the results, it is observed that the ToxicXLMR based multitask model improves accuracy by 4% compared to the STL-C model. It is also observed that the ToxicXLMR based models performed better than the RoBERTa and

[13] https://huggingface.co/

BERT based models and by applying the multi-task models, there is at least a 1% improvement in classification accuracy compared to the STL-C models across the transformer variants. The classification performance of BERT-MT (Xiang et al., 2021) is comparable to the proposed model using BERT and ToxicBERT transformers. But, the proposed model with XLMR transformer has significantly better accuracy and F1 score (4%) than this BERT-MT.

6.2 Span prediction performance

To evaluate our models for the toxic span prediction task (rationale extraction), the TSD (SemEval-2021) dataset was used. The ad-hoc assessment measure proposed by Da San Martino et al. (2019) was used to evaluate span prediction performance. This was officially used to rank SemEval-2021 Task 5 submissions. The ad-hoc assessment metric provides partial credit to incomplete character matches. For a document d, S_d is the set of toxic character offsets predicted by the system and G_d is the set of ground truth annotations. Then the F_1^d score of the system for document d is defined as

$$F_1^d = \frac{2 * P^d * R^d}{P^d + R^d}, \text{ where } P^d = \frac{|S_d \cap G_d|}{|S_d|} \text{ and } R^d = \frac{|S_d \cap G_d|}{|G_d|} \tag{9}$$

When a document does not have a ground truth annotation ($G_d = \varnothing$), or the system does not output character offset prediction ($S_d = \varnothing$), then (10) is used.

$$F_1^d = \begin{cases} 1 \text{ when } G_d = S_d = \emptyset, \\ 0 \text{ otherwise} \end{cases} \tag{10}$$

Finally, the average score of F_1 on all test samples of an evaluation dataset is the model F1 score.

The F1 score of the proposed multi-task model is on par with the best ensemble models of the leader board for SemEval-2021 Task 5 (Pavlopoulos et al., 2021). As shown in Table 5, its F1 score is 69.38 while the top two ensemble models' F1 scores are 70.83 and 70.77, respectively. HITSZ-HLT (Zhu et al., 2021) and S-NLP (Nguyen et al., 2021) are the ensembles of three transformer-based models which has 3x inference time compared to the proposed MTL model. Benchmark 1 (Nguyen et al., 2021) in Pavlopoulos et al. (2021) used a fine-tuned version of RoBERTa and achieved a competitive F1 score, but can only identify the toxic span and is not applicable for the classification of toxic comments. One requires to create a pipeline of toxic comment classification and toxic span prediction models to identify both the label and toxic span of the input sequence for Benchmark 1. Whereas the proposed single model can do both tasks with a competitive F1 score for toxic span prediction and toxic comment classification. BERT-MT (Xiang et al., 2021) model's span prediction score is very low compared to ToxicXLMR-MTL model as shown in Table 5. This result shows the significance of Bi-LSTM CRF model for span prediction compared to the individual token classification as done in Xiang et al. (2021).

6.3 Generalization ability of the model

We are interested in determining the impact of the coverage of captured phenomena by the models. A classifier trained on a larger coverage dataset and tested on a smaller

Table 4 Evaluation Results for Toxic Span Prediction task on TSD test dataset

Transformer / Model	MTL F1 Score	STL-SP F1 Score
XLMR	68.35 ± 0.14	67.62 ± 0.31
ToxicXLMR	**69.38 ± 0.13**	**67.82 ± 0.15**
RoBERTa	66.34 ± 0.32	66.44 ± 0.53
ToxicRoBERTa	66.51 ± 0.15	66.68 ± 0.40
BERT	62.99 ± 0.24	60.43 ± 0.74
ToxicBERT	64.88 ± 0.21	64.49 ± 0.44

The F1 score gives credits to the partial predictions. The best performance is in bold. Each model is run three times with different seed values and their average is reported here

coverage dataset will give a good performance according to Pamungkas and Patti (2019). Hence, we tested the proposed MTL model on HASOC and OLID datasets which are not exposed to the model during training. We performed the Mann-Whitney U test (Nachar, 2008) on these three datasets to find the overlap of word distributions. We found that 30% of the words of curated and HASOC datasets, and 23% of curated and OLID datasets are having different distributions with confidence of 95%. This shows the dissimilarity of curated, HASOC and OLID datasets. From the results shown in Table 6, it can be observed that MTL models have better results over STL-C models. We have observed 3% and 6% improvement in weighted F1 scores of all the MTL variants compared to the STL-C variants on HASOC and OLID datasets, respectively. The best published weighted F1 score on the HASOC dataset is 83.95 while our best multi-task model's (ToxicXLMR-MTL) weighted F1 score is 80.76 (Wang et al., 2019). Similarly, the best published macro F1 score on the OLID dataset is 83 (Liu et al., 2019a) and our best multi-task model's score is 77.53.

7 Discussion

This section provides a discussion about suitable pretrained language models, the effect of their fine-tuned versions, the performance impact of the MTL models compared to STL models, and the error analysis of the top-performing model.

Table 5 Comparing our model with best performing models on the TSD dataset

Model	F1 Score
HITSZ-HLT (Zhu et al., 2021)	70.83
S-NLP (Nguyen et al., 2021)	70.77
Benchmark 1 (Pavlopoulos et al., 2021)	69.89
‡BERT-MT (Xiang et al., 2021)	58.93
ToxicXLMR-MTL (ours)	69.38

‡ Replicated model

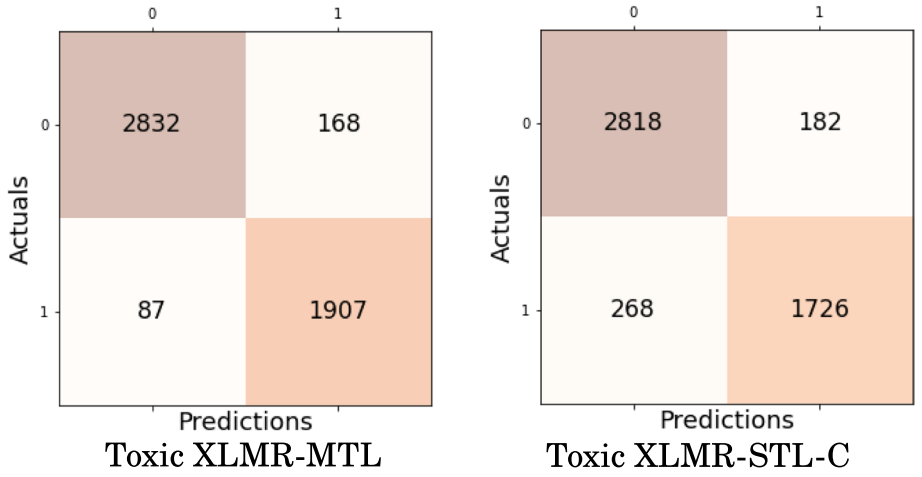

Fig. 5 **left:** Confusion matrix for multi-task model, **right:** Confusion matrix for classification task alone model; in the figure, 0 means non-toxic and 1 means toxic

Table 6 Model Evaluation Results on both HASOC and OLID for domain adaptation

Model	Results on HASOC 2019 English-Test dataset			Results on OLID 2019 English-Test dataset		
	Accuracy	Weighted F1 Score	Macro F1 Score	Accuracy	Weighted F1 Score	Macro F1 Score
Multi-task models (MTL)						
XLMR	79.18 ± 0.13	79.67 ± 0.18	73.58 ± 0.16	81.28 ± 0.15	80.54 ± 0.17	76.00 ± 0.17
ToxicXLMR	**80.40 ± 0.15**	**80.77 ± 0.16**	**74.87 ± 0.15**	**83.26 ± 0.12**	**82.54 ± 0.16**	**77.53 ± 0.14**
RoBERTa	79.97 ± 0.11	80.35 ± 0.16	74.34 ± 0.13	81.98 ± 0.13	81.49 ± 0.18	76.44 ± 0.24
ToxicRoBERTa	79.79 ± 0.19	80.16 ± 0.19	74.06 ± 0.20	82.09 ± 0.19	81.40 ± 0.17	76.12 ± 0.14
BERT	74.50 ± 0.23	75.56 ± 0.18	69.07 ± 0.20	75.93 ± 0.12	76.30 ± 0.22	71.05 ± 0.18
ToxicBERT	74.59 ± 0.16	75.90 ± 0.13	70.12 ± 0.14	73.95 ± 0.15	74.32 ± 0.17	68.57 ± 0.12
‡BERT-MT (Xiang et al., 2021)	74.52 ± 0.21	74.54 ± 0.22	69.02 ± 0.27	74.28 ± 0.51	74.57 ± 0.47	69.52 ± 0.42
Classification task models (STL-C)						
XLMR	77.19 ± 0.19	77.52 ± 0.20	70.47 ± 0.20	76.16 ± 0.18	75.17 ± 0.23	68.07 ± 0.21
ToxicXLMR	**78.06 ± 0.16**	**78.42 ± 0.15**	**71.71 ± 0.13**	**77.33 ± 0.14**	**76.47 ± 0.22**	**69.81 ± 0.15**
RoBERTa	72.33 ± 0.14	73.72 ± 0.21	67.31 ± 0.15	71.28 ± 0.16	71.78 ± 0.17	65.62 ± 0.21
ToxicRoBERTa	73.37 ± 0.12	74.58 ± 0.15	68.07 ± 0.10	72.44 ± 0.19	72.98 ± 0.20	67.16 ± 0.18
BERT	72.51 ± 0.21	73.72 ± 0.22	66.92 ± 0.15	70.23 ± 0.20	71.01 ± 0.24	65.09 ± 0.11
ToxicBERT	73.20 ± 0.13	74.57 ± 0.14	68.43 ± 0.10	71.40 ± 0.21	72.19 ± 0.18	66.60 ± 0.13

‡ Replicated model

We report accuracy, weighted and macro F1 scores. The best performance is bold. Each model is run three times with different seed values and their average is reported here

7.1 Influence of pretrained language models

We conducted extensive experiments with several pretrained language models in order to analyze the performance of the shared parameter mechanism used in the proposed MTL model. From the results, first, we show that integrating pretrained language models with MTL improves the effectiveness of toxic span prediction and toxic comment classification. The pretrained model utilized 12 transformer layers constructed from a multi-head attention network, which assist in extracting contextual information of tokens. The MTL allows the system to share knowledge between tasks and exploits knowledge from TSP task to boost TCC task. From the results (Tables 3 and 4), it can be observed that the type of the transformer model and the corpus utilized to pre-train and fine-tune also effects the performance of the proposed model. The fine-tuned versions of XLMR and RoBERTa showed improvement compared to their base versions. For example, ToxicXLMR-MTL has shown a 1.8% and 1% improvement over XLMR-MTL for TCC and TSP tasks, respectively, in terms of macro F1 score. It can also be seen that from Table 6, ToxicXLMR improved the domain adaptation performance by 6% and 10% compared to BERT when tested on HASOC and OLID test sets, respectively, in terms of accuracy and F1 score.

7.2 Performance evaluation with baselines

We compared the performance of the proposed MTL system with two baseline models built on the single task learning (STL) methodology. The results suggest that MTL outperformed the STL baseline model in terms of accuracy and F1 score. For instance, MTL (ToxicXLMR-MTL) model shown 4% and 1.7% improvement in F1 scores compared to STL (ToxicXLMR) models for TCC and TSP tasks, respectively. ToxicXLMR-MTL has shown 9% improvement for toxic span prediction task compared to BERT-MT (Xiang et al., 2021). When the models are evaluated on an unseen dataset to test for domain adaptation ability, the MTL model (ToxicXLMR-MTL) has shown 3% and 8% improvement in macro F1 score on HASOC and OLID, respectively, compared to the STL model (ToxicXLMR-STL-C). We also compared the proposed MTL model with the top-ranking ensembles of SemEval-2021 task 5 for the TSP task. The MTL model (ToxicXLMR-MTL) has shown a competitive F1 score (Table 5).

7.3 Error analysis

In spite of performance, the proposed MTL model is unable to handle a few errors. In these sections, we examine the errors of the best-performing model (ToxicXLMR-MTL).

7.3.1 Classification analysis

Figure 5 shows the confusion matrices of the predicted class labels against the ground truth class labels for both ToxicXLMR-MTL and ToxicXLMR-STL-C models (these two models are performing best, can be seen in Tables 3 and 6). From Fig. 5 it can be observed that false negatives (toxic comments predicted as non-toxic) and false positives are high in the ToxicXLMR-STL-C compared to the ToxicXLMR-MTL.

7.3.2 Analysis of false positives

A qualitative analysis of the results of the ToxicXLMR-MTL model revealed that false positives (non-toxic predicted as toxic) are due to negative words such as racist, screw, traitor, stupid, troll, etc. The following five comments (*a* to *e*) shown below are actually non-toxic. However, the ToxicXLMR-MTL model predicted them as toxic due to the presence of negative words. The toxic spans identified by the model are shown in bold. For example, comment (*b*) is labelled as toxic due to the presence of the word "*pussy*" (a negative word). We are able to identify the cause for this misclassification due to the identified toxic span by the model. The mere presence of negative words may not make a comment as toxic. From Fig. 5, we can observe that the MTL model has low false positives (168) when compared to the STL model (182) for the curated dataset. The BiLSTM CRF layer considers the sequence of text in determining the toxic span. The use of joint loss which gives equal importance to both classification and span prediction tasks have a positive influence on results of both the tasks. However, the importance of negative words as learned by the model dominates the non-toxic usage of toxic words.

a) *Sounds like* racism *and xenophobia to me. I personally am not afraid of brown and black people but I am completely nauseated by people like you*
b) *and that's what you got out of the whole thing - you say pink* pussy*(cat) hats - what are you.*
c) *morally* disgusting
d) *Getting* screwed *by a bank. who would have thought*
e) *The Lynch comment refers to a bout of renaming in the Centennial School District. We also have another school district in another state renaming a school from Jefferson Davis High to the name of another* traitor, *Barry Obama*

The false positives and false negatives occur due to mere presence of doubtful labels, toxicity without swear/profane words, sarcasm, irony, and usage of swear words in non-toxic context (Van Aken et al., 2018). We wish to study the effect of semantic embeddings to address these challenges in the future.

7.3.3 Analysis of false negatives

The major source of false negatives (toxic comments predicted as non-toxic) can be attributed to lack of context. As machines are unaware of the context of a particular sentence used and indirect toxicity, it is hard to predict that the sentence is toxic, even though it is toxic. Sometimes it is even hard for humans to figure out such implicit toxicity without the knowledge of the context of the discussion. The current models are far from the human understanding of language.

The following four are the examples for toxic comments labelled as non-toxic by ToxicXLMR-MTL.

a) *I do. My father shot at them and I'd be happy to follow in his footsteps. Waste of good lead, though*
b) *I agree. I think we should stop calling them "alt-right" and start calling them Nazis instead*
c) *Ignorance must be bliss for you*

d) *Good job smelly. way to attack the message and not the messenger*

7.3.4 Span prediction analysis

An analysis of the predicted spans by ToxicXLMR-MTL and ground truth toxic spans is provided here, with example comments. Ground truth spans are bold-faced, and model predicted spans are underlined. For these example comments, the ToxicXLMR-STL-SP model predicted an empty span. From these examples, we can observe that the MTL model is able to identify subtle toxicity based on the context to some extent, though the model doesn't actually understand the text as humans do. It can be observed that comment *a* is toxic due to the word *depraved*, but the human annotator used a different word *homosexual*. This disparity is due to the difference in human annotator's opinion in tagging the toxic span.

a) *Yes they would, then hey could oh so self righteous about how depraved all* homosexuals *are and this just proves it!* (Prediction matches the ground truth.)
b) *The 2nd Amendment is a white privilege*. Black folks are shot on the mere suspicion of being armed. *That includes black children* (Predicted span is a subset of ground truth.)
c) *You're as full of* excrement *as Trump, your malignant narcissist hero* (There is a mismatch between predicted and ground truth spans. As both of these words are actually toxic, the model might have picked one among them.)
d) *How much longer does this country have to suffer this* disgusting moron *as president* (Predicted span is subset of ground truth.)
e) *Blah, Blah, Blah. The Obama administration* sucked, *end of story* (Prediction matches the ground truth.)

8 Conclusion and future work

This paper demonstrated a multi-task model that performs toxic comment classification while predicting the toxic spans as rationales. We have curated a dataset containing both class label and toxic span information to train the MTL model, as no such data set exists as of date. The multi-task model is built using a transformer (XLMR) based Bi-LSTM CRF architecture. The proposed model has better classification and span prediction performance, as the model uses joint loss function over the related tasks. The model exhibited a competitive F1 score on the SemEval-2021 Task 5 dataset for the toxic span prediction task when compared to the models in the leaderboard. The proposed model has shown a 4% improvement in F1 score for the classification task when compared to the STL model. It is observed that the type of the transformer model and the corpus utilized to pre-train and fine-tune also effects the performance of the proposed model. The empirical evidence of testing on out of domain datasets HASOC and OLID shows that the proposed model is effective for both in-domain and out-domain evaluation. In the future, we intend to develop models using semantic embeddings that take more delicate context and actors of text into account in order to handle the subtle differences in the usage of toxic keywords.

Author Contributions All authors have equal contribution.

Data Availability The datasets used and/or analyzed during the current study will be available from the corresponding author upon request.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for Publication Not applicable.

Competing interests The authors declare that they have no competing interests.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, *6*, 52,138–52,160. https://doi.org/10.1109/ACCESS.2018.2870052

Ahmed, SS, & Kumar M., A. (2021). Classification of censored tweets in Chinese language using XLNet. In *Proceedings of the fourth workshop on NLP for internet freedom: Censorship, disinformation, and propaganda*. https://doi.org/10.18653/v1/2021.nlp4if-1.21 (pp. 136–139).

Akhtar, M.S., Garg, T., & Ekbal, A. (2020). Multi-task learning for aspect term extraction and aspect sentiment classification. *Neurocomputing*, *398*, 247–256. https://doi.org/10.1016/j.neucom.2020.02.093

Ashok Kumar, J, Abirami, S, Tina Esther, T., & et al (2021). Comment toxicity detection via a multichannel convolutional bidirectional gated recurrent unit. *Neurocomputing*, *441*, 272–278. https://doi.org/10.1016/j.neucom.2021.02.023

Badjatiya, P., Gupta, S., Gupta, M., & et al (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on world wide web companion*. https://doi.org/10.1145/3041021.3054223 (pp. 759–760).

Bansal, A., Kaushik, A., & Modi, A. (2021). IITK@detox at SemEval-2021 task 5: Semi-supervised learning and dice loss for toxic spans detection. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*. https://doi.org/10.18653/v1/2021.semeval-1.24 (pp. 211–219).

Baxter, J. (2000). A model of inductive bias learning. *Journal of artificial intelligence research*, *12*, 149–198. https://doi.org/10.5555/1622248.1622254

Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41–75. https://doi.org/10.1023/A:1007379606734

Caruana, R.A (1993). Multitask connectionist learning. In *Proceedings of the 1993 connectionist models summer school*.

Caselli, T., Basile, V., Mitrović, J., & et al (2021). HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th workshop on online abuse and harms (WOAH 2021)*. https://doi.org/10.18653/v1/2021.woah-1.3(pp. 17–25).

Chakrabarty, T., Gupta, K., & Muresan, S. (2019). Pay "attention" to your context when classifying abusive language. In *Proceedings of the third workshop on abusive language online*. https://doi.org/10.18653/v1/W19-3508 (pp. 70–79).

Chen, Q., Zhuo, Z., & Wang, W. (2019). Bert for joint intent classification and slot filling. arXiv:1902.10909

Chia, Z.L., Ptaszynski, M., Masui, F., & et al (2021). Machine learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection. *Information Processing and Management*, *58*, 102600. https://doi.org/10.1016/j.ipm.2021.102600

Conneau, A., Khandelwal, K., Goyal, N., & et al (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. https://doi.org/10.18653/v1/2020.acl-main.747 (pp. 8440–8451).

Da San Martino, G., Yu, S., Barrón-Cedeño, A., & et al (2019). Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. https://doi.org/10.18653/v1/D19-1565 (pp. 5636–5646).

Davidson, T., Warmsley, D., Macy, M., & et al (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*. https://ojs.aaai.org/index.php/ICWSM/article/view/14955 (pp. 512–515).

Dellerman, D. (2022). Influence of cyberbullying on suicidal behaviors. Ph.D. Thesis, Walden University.

Devlin, J., Chang, M.-W., Lee, K., & et al (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (Long and Short Papers)*. https://doi.org/10.18653/v1/N19-1423 (pp. 4171–4186).

Ed-drissiya, E, Sarrouti, M., En-Nahnahi, N., & et al (2021). Mttlade: A multi-task transfer learning-based method for adverse drug events extraction. *Information Processing and Management*, *58*(3), 102473. https://doi.org/10.1016/j.ipm.2020.102473

Elnaggar, A., Waltl, B., Glaser, I., & et al (2018). Stop illegal comments: A multi-task deep learning approach. In *Proceedings of the 2018 artificial intelligence and cloud computing conference*. https://doi.org/10.1145/3299819.3299845 (pp. 41–47).

Fortuna, P., Soler-Company, J., & Wanner, L. (2021). How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Information Processing and Management*, *58*(3), 102524. https://doi.org/10.1016/j.ipm.2021.102524

Founta, A.M., Chatzakou, D., Kourtellis, N., & et al (2019). A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM conference on web science*. https://doi.org/10.1145/3292522.3326028 (pp. 105–114).

Gong, T., Lee, T., Stephenson, C., & et al (2019). A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, *7*, 141627–141632. https://doi.org/10.1109/ACCESS.2019.2943604

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. arXiv:1508.01991

Karen, S., Andrea, V., & Andrew, Z. (2014). Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv:1312.6034

Khan, Y., Ma, W., & Vosoughi, S. (2021). Lone pine at SemEval-2021 task 5: fine-grained detection of hate speech using BERToxic. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*. https://doi.org/10.18653/v1/2021.semeval-1.132 (pp. 967–973).

Kiran Babu, N., & HimaBindu, K. (2022). Attention-based bi-lstm network for abusive language detection. *IETE Journal of Research*, 1–9. https://doi.org/10.1080/03772063.2022.2034534

Liu, P., Li, W., & Zou, L. (2019a). NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th international workshop on semantic evaluation*. https://doi.org/10.18653/v1/S19-2011 (pp. 87–91).

Liu, Y., Ott, M., Goyal, N., & et al. (2019b). Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692

Long, M., Cao, Z., Wang, J., & et al (2017). Learning multiple tasks with multilinear relationship networks. In *Proceedings of the 31st international conference on neural information processing systems*. https://dl.acm.org/doi/10.5555/3294771.3294923(pp. 1593–1602).

Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*. https://doi.org/10.18653/v1/P16-1101(pp. 1064–1074).

Mathew, B., Saha, P., Yimam, SM, & et al (2021). Hatexplain: a benchmark dataset for explainable hate speech detection, 14,867–14,875. https://ojs.aaai.org/index.php/AAAI/article/view/17745

McCann, B., Keskar, N.S., Xiong, C., & et al. (2018). The natural language decathlon: Multitask learning as question answering. arXiv:1806.08730

Mehta, D., Dwivedi, A., Patra, A., & et al (2021). A transformer-based architecture for fake news classification. *Social Network Analysis and Mining*, *11*(1), 1–12. https://doi.org/10.1007/s13278-021-00738-y

Mozafari, M., Farahbakhsh, R., & Crespi, N. (2019). A BERT-based transfer learning approach for hate speech detection in online social media. In *Complex networks 2019: 8th international conference on complex networks and their applications*. https://doi.org/10.1007/978-3-030-36687-277 (pp. 928–940).

Nachar, N. (2008). The mann-whitney u: A test for assessing whether two independent samples come from the same distribution. *Tutorials in Quantitative Methods for Psychology*, *4*, 13–20. https://doi.org/10.20982/tqmp.04.1.p013

Nguyen, VA, Nguyen, TM, Quang Dao, H., & et al (2021). S-NLP at SemEval-2021 task 5: An analysis of dual networks for sequence tagging. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*. https://doi.org/10.18653/v1/2021.semeval-1.120 (pp. 888–897).

Pamungkas, EW, & Patti, V. (2019). Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual*

meeting of the association for computational linguistics: student research workshop. https://doi.org/10.18653/v1/P19-2051 (pp. 363–370).

Park, JH, & Fung, P. (2017). One-step and two-step classification for abusive language detection on Twitter. In *Proceedings of the first workshop on abusive language online*. https://doi.org/10.18653/v1/W17-3006 (pp. 41–45). Vancouver: Association for Computational Linguistics.

Pavlopoulos, J., Sorensen, J., Laugier, L, & et al (2021). SemEval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*. https://doi.org/10.18653/v1/2021.semeval-1.6 (pp. 59–69).

Ramsundar, B., Kearnes, S., Riley, P., & et al. (2015). Massively multitask networks for drug discovery. arXiv:1502.02072

Ribeiro, MT, Singh, S., & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*. https://doi.org/10.1145/2939672.2939778 (pp. 1135–1144).

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. arXiv:1706.05098

Schuster, M., & Nakajima, K. (2012). Japanese and korean voice search. In *2012 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. https://doi.org/10.1109/ICASSP.2012.6289079 (pp. 5149–5152).

Sharma, M., Kandasamy, I., & Vasantha, W.b. (2021). YoungSheldon at SemEval-2021 task 5: Fine-tuning pre-trained language models for toxic spans detection using token classification objective. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*. https://doi.org/10.18653/v1/2021.semeval-1.130(pp. 953–959).

Sonone, S. S, Sankhla, MS, & Kumar, R. (2021). Cyber bullying. In *Combating the exploitation of children in cyberspace: emerging research and opportunities*. https://doi.org/10.4018/978-1-7998-2360-5.ch001 (pp. 1–18).

Standley, T., Zamir, A., Chen, D., & et al (2020). Which tasks should be learned together in multi-task learning?. In *Proceedings of the 37th international conference on machine learning*. https://proceedings.mlr.press/v119/standley20a.html(pp. 9120–9132).

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proceedings of the 34th international conference on machine learning*. https://dl.acm.org/doi/10.5555/3305890.3306024 (pp. 3319–3328).

Temper, M., Poisel, R., & Tjoa, S. (2013). Facebook watchdog: A research agenda for detecting online grooming and bullying activities. In *IEEE International conference on systems, man, and cybernetics, SMC*. https://doi.org/10.1109/SMC.2013.487 (pp. 2854–2859).

Van Aken, B., Risch, J., Krestel, R., & et al (2018). Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*. https://doi.org/10.18653/v1/W18-5105 (pp. 33–42).

Vaswani, A., Shazeer, N., Parmar, N., & et al (2017). Attention is all you need. In *Advances in neural information processing systems*. https://dl.acm.org/doi/10.5555/3295222.3295349

Viterbi, A. J. (2009). Viterbi algorithm. *Scholarpedia*, *4*(1), 6246. https://doi.org/10.4249/scholarpedia.6246

Wang, B., Ding, Y., Liu, S., & Zhou, X. (2019). Ynu_wb at HASOC 2019: Ordered neurons LSTM with attention for identifying hate speech and offensive language. In *Working notes of FIRE 2019 - forum for information retrieval evaluation*. http://ceur-ws.org/Vol-2517/T3-2.pdf (pp. 191–198).

Wang, X., Xu, G., Zhang, Z., & et al (2021). End-to-end aspect-based sentiment analysis with hierarchical multi-task learning. *Neurocomputing*, *455*, 178–188. https://doi.org/10.1016/j.neucom.2021.03.100

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL student research workshop*. https://doi.org/10.18653/v1/N16-2013 (pp. 88–93).

Wiegreffe, S., & Pinter, Y. (2019). Attention is not not explanation. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. https://doi.org/10.18653/v1/D19-1002 (pp. 11–20).

Worsham, J., & Kalita, J. (2020). Multi-task learning for natural language processing in the 2020s: where are we going? *Pattern Recognition Letters*, *136*, 120–126. https://doi.org/10.1016/j.patrec.2020.05.031

Xiang, T., Macavaney, S., Yang, E., & et al (2021). Toxccin: Toxic content classification with interpretability. In *Proceedings of the 11th workshop on computational approaches to subjectivity, sentiment and social media analysis*. https://aclanthology.org/2021.wassa-1.1 (pp. 1–12).

Xu, K., Ba, J., Kiros, R., & et al (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd international conference on machine learning*. https://doi.org/10.5555/3045118.3045336(pp. 2048–2057).

Zaidan, O., Eisner, J., & Piatko, C. (2007). Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: the conference of the North American*

chapter of the association for computational linguistics; proceedings of the main conference. https://aclanthology.org/N07-1033 (pp. 260–267).

Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*. https://doi.org/10.1007/978-3-319-93417-4_48 (pp. 745–760).

Zhu, Q., Lin, Z., Zhang, Y., & et al (2021). HITSZ-HLT at SemEval-2021 task 5: Ensemble sequence labeling and span boundary detection for toxic span detection. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*. https://doi.org/10.18653/v1/2021.semeval-1.63 (pp. 521–526).

Zou, L., & Li, W. (2021). LZ1904 at SemEval-2021 task 5: Bi-LSTM-CRF for toxic span detection using pretrained word embedding. In *Proceedings of the 15th international workshop on semantic evaluation (SemEval-2021)*. https://doi.org/10.18653/v1/2021.semeval-1.138 (pp. 1009–1014).