Knowledge-Based Systems 210 (2020) 106458

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A deep neural network based multi-task learning approach to hate speech detection

Prashant Kapil *, Asif Ekbal

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Bihar, India

ARTICLE INFO

Article history:
Received 18 April 2020
Received in revised form 9 August 2020
Accepted 2 September 2020
Available online 6 October 2020

Keywords:
Multi-task learning
Hate speech detection
Shared features
Task specific features
Macro-F1
Weighted-F1

ABSTRACT

With the advent of the internet and numerous social media platforms, citizens now have enormous opportunities to express and share their opinions on various societal and political issues. This phenomenal growth of the internet, social media networks, and messaging platforms provide plenty of opportunities for building intelligent systems, but these are also being heavily misused by certain groups who often disseminate offensive, racial, and hate speeches. Hence, detecting hate speech at the right time plays a crucial role as its spread might affect social fabrics. In recent times, although a few benchmark datasets have emerged for hate speech detection, these are limited in volume and also do not follow any uniform annotation schema. In this paper, a deep multi-task learning (MTL) framework is proposed to leverage useful information from multiple related classification tasks in order to improve the performance of the individual task. The proposed multi-task model is based on the shared-private scheme that assigns shared and private layers to capture the shared-features and task-specific features from five classification tasks. Experiments[1] on the 5 datasets show that the proposed framework attains encouraging performance in terms of macro-F1 and weighted-F1.

1. Introduction

With the phenomenal growth in digital technology and the internet, social media have upsurged as a strong platform to allow people to express their opinions on a variety of topics ranging from political, financial, education, sports, defense, religion and other societal issues. Statistics reveals that 6K tweets/second, 200 billion tweets/year[2] are generated on twitter alone, indicating the exponential rise in the consumption of social media. The diversity in language usage across the globe also poses a great challenge due to the variety of linguistic patterns. Social media's main aim is to connect more people to support them in expressing their right to freedom of speech.

However, these mediums are often misused by certain groups to malign others, spreading offensive and hate speeches targeting individuals and/or other groups. This can be considered as a political violence that jeopardizes social stability and peace. Hence, detecting these in proper time and preventing their dissemination to a larger section is of utmost importance to maintain the harmony in the society and to maintain the law and order situations.

United Nations strategy and plan of action on hate speech describes hate speech as *any kind of communication in speech, writing or behavior, that attacks or uses pejorative or discriminatory language concerning a person or a group based on who they are, in other words, based on their religion, ethnicity, nationality, race, color, descent, gender or identity factor*. So without suppressing the right to freedom of expression, the focus should be on building robust computational systems which can detect different types of hateful contents that can create disharmony. *International Covenant on Civil and Political Rights*[3] (ICCPR) is a multilateral treaty adopted by United Nations General Assembly. The covenant commits its party to respect the right to *freedom of speech* along with other fundamental rights for every citizen. As of September 2019, the covenant has 173 parties. *Article 19* of it states that:

(i). Everyone shall have the right to hold opinions without interference.

(ii). Everyone shall have the right to freedom of expression. This right shall include freedom to seek, receive and impart information of all kinds, regardless of frontiers, either orally, in writing or print, in the form of art, or through any media of person's choice.

However, an amendment was done and a new article 20 was introduced stating that any advocacy of national, religious or

* Corresponding author.
E-mail addresses: prashant.pcs17@iitp.ac.in (P. Kapil), asif@iitp.ac.in (A. Ekbal).

1 Code is available at https://github.com/imprasshant/MTL.

2 https://www.internetlivestats.com/twitter-statistics/.

3 https://en.wikipedia.org/wiki/International_Covenant_on_Civil_and_Political_Rights.

https://doi.org/10.1016/j.knosys.2020.106458
0950-7051/

racial hatred that constitutes incitement to discrimination, hostility, violence shall be prohibited under law. There are incidents when viciousness of these messages evolved into genocide, xenophobia and bigotry. Several incidents in the past had evidence of deadly action like mass murder before posting hate messages in online forums. [1] reported the massive violence in Kenya after hateful messages circulated in media in 2007–2008. In order to build efficient machine learning based hate speech detection system, sufficient amount of labeled data is required. Although there exists a few benchmark datasets, they are often limited by the size, and do not follow any uniform annotation schema.

In Table 1, various types of datasets are shown which have been collected from the various online forums, blogs and social media platforms. The laws of some countries to deal with hate speech is given in Table 2. Based on the definitions of hate speech in Table 1 it is quite clear that irrespective of geographical location and diverse culture there are often overlapped concepts while defining hate. In this paper we propose a deep learning based end-to-end multi-tasking approach to leverage the information from multiple related tasks, *viz.* hate, racism, sexism and offensive contents etc. The major contributions of this paper are summarized below:

1. We propose a Shared-Private Multi-Task Learning (SP-MTL) framework to leverage the benefits of multiple related tasks, such as hate speech classification, offensive language identification, racism detection and sexism detection etc. The SP-MTL model introduces two feature spaces for each task: one is to store the shared features among these related tasks by training in a jointly manner, and the other is to capture the task dependent features. To the best of our knowledge this is the very first attempt towards building an end-to-end deep neural network based multi-tasking framework for hate speech detection.
2. The shared knowledge learned by SP-MTL (explained in Section 3.5) model can be considered as off-the-shelf-knowledge and can be transferred to the new task relevant to hate speech.
3. Efficacy of the proposed model is demonstrated through the detailed empirical evaluation results on five benchmark datasets.

The remaining structure of this paper is as follows. A brief overview of the related background literature is presented in Section 2. Section 3 discusses in details the proposed methodology. In Section 4, the datasets used for the experiments and definitions of different variants of hate are described. Experimental setup and evaluation metrics are presented in Section 5. Section 6 reports the evaluation results and comparisons to the state-of-the-arts. Error analysis containing qualitative and quantitative analysis of the obtained results are presented in Section 7. Finally, the conclusion and directions for future research are presented in Section 8.

2. Related work

Most of the previous works carried out in this direction have mainly focused on supervised classification of different subtypes of hate. The problem has mostly been modeled concerning a single-task learning framework. Single Task Learning (STL) is a paradigm that updates the weight of the neural networks using input sequence from only one classification task involving one dataset. Different types of classification algorithms have been utilized: feature based traditional machine learning techniques, and deep neural network based techniques that do not require any handcrafting of features. The features used are diverse in nature, varying from lexical to syntactic to semantics levels. In this section, various feature types used in the supervised setting and other techniques suggested to improve the classifier performance is discussed.

2.1. Surface features and word embedding features

[7] sampled data from the comments posted on Yahoo! News, and finance. They build a supervised classification methodology using Vowpal Wabbit's regression model[4] utilizing features such as character n-grams, token unigrams, and bigrams, word2vec features, etc. [4] made public an annotated corpus of ≈ 16K tweets. They evaluated logistic regression classifier utilizing char n-grams and word n-grams based features and found that character n-grams outperform word n-grams, due to character n-gram matrices being far less sparse than word n-gram matrices. [8] investigated deep neural networks, namely Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) by initializing word embeddings with random embedding, FastText word embeddings [9] and GloVe word embeddings [10] using data by [4]. [11] constrained their work to binary classification between *abusive* and *not abusive*. Their character-based approach outperformed token-based and distributional-based features on the dataset by [7].

[12] used surface features such as word unigram, bigram, and trigram each weighted by term frequency–inverse document frequency (TF–IDF), number of characters, number of words and hashtags fed into support vector machine (SVM). Their best approach of CNN stacked Gated Recurrent Unit (GRU) leverages google pre-trained word2vec [13] and achieved robust F1 score for six different datasets. They concluded that the presence of abstract concepts, such as *racism*, *sexism* and *hate* are very difficult to detect solely based on textual content. [14] trained four CNN models, based on character 4-grams, word vectors based on semantic information built using word2vec, randomly generated word vectors, and word vectors combined with character n-grams utilizing the dataset by [15]. [16] utilized BERT by [17] on the dataset by [18,19] considered Bidirectional-LSTM (BiLSTM) with a dense layer on top consuming ELMo vectors [20], and Bag-of-words for classifying hate and non-hate on the dataset by [21]

2.2. Lexical resources

In some of the prior works, lexical resources have also been used. For example, domain-specific dictionaries containing abusive words and negative words indicative of aggressive behavior have been utilized to extract useful features and to be used in the supervised machine learning model. [22] generated a lexicon of sentiment expressions using semantic and subjectivity features with an orientation to hate speech to be used in developing a rule-based classifier. Beginning with an initial seed list of six verbs, namely *discriminate*, *loot*, *riot*, *beat*, *kill* and *evict*, they used concept of bootstrapping, WordNet's synsets and hypernym relationship to extend the lexicons. [23] collected 2700 words, phrases, and expressions with different degrees of manifestation of flame varieties. Each entry is assigned a weight in the range of 1–5 based on the potential impact level of each entry in the dissemination of hate posts. This collection is termed as Insulting Abusive Language Dictionary (IALD). They utilized two different sources of data to build a three-level classifier using Complement Naive Bayes classifier and Multinomial updatable Naive Bayes classifier in WEKA [24] as first two levels. The outputs of this classification level are new aggregated features extracted from the previous level feature space, with the following attributes as the input for the last-level classification task, using IALD: frequency of IALD words, the maximum weight of IALD entries, the normalized average weight of IALD entries, the probability that the current instance is okay or flame based on previous

[4] https://github.com/JohnLangford/vowpal_wabbit.

Table 1
Definition of hate to collect data.

Authors	Definition
[2]	a language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate or insult the members of the group.
[3]	a speech that denigrates any person or any group based on characteristics like race, color, gender, religion, ethnicity, nationality, sexual preferences etc.
[4]	A tweet is offensive if it contains racist or sexist slur, intention to attack, promote violent crimes, threatening minorities, and stereotyping genders.
[5]	It is a bias-motivated hostile speech aimed at a person or group of people with intentions to injure, dehumanize, harass, degrade and victimizing targeted groups based on some innate characteristics.
[6]	It is defined as abusive speech containing a high frequency of stereotypical words.

Table 2
Countries law on hate speech.

Country	Law
USA	Hate speech is legally protected free speech under the First Amendment. However, speech that include obscenity,speech integral to illegal conduct, speech that incites lawless action or likely to produce such activity are given lesser or no protection.
Brazil	According to the 1988 Brazilian constitution racism is an offense with no statute of limitations and no right to bail for the defendant.
Germany	Section 130 of Germany criminal code states incitement to hatred is a punishable offense leading up to 5 years imprisonment. It also states that publicly inciting hate against some parts of population or using insulting malicious slur or defaming to violate their human dignity is a crime.
India	Article 19(1) of the constitution of India protects the freedom of speech and expression. However, article 19(2) states that to protect sovereignty, integrity, and security of the state, to protect decency and morality,defamation and incitement to an event, some restriction can be imposed.
Japan	The Hate speech act of 2016 does not apply to groups of people but covers threats and slander to protect.
New Zealand	Their Hate speech act follows Section 61 of the Human Rights Act 1993 that asserts that threatening, abusive contents in any form, words that are likely to create hostility against a group of people on the basis of race, color, ethnicity is unlawful.
Russia	Article 282 of the Criminal Code asserts that inciting hatred or antagonism, disparaging a person or the group of people based on sex, race, nationality, language, origin, affiliation to any social group is punishable with fine or imprisonment up to 2 years or have to undergo obligatory, corrective or compulsory labour.
France	Its principal hate speech legislation is Press law of 1881, in which Section 24 criminalizes incitement to racial discrimination, hatred, or violence on the basis of one's origin or membership in an ethnic, national, racial or religious group.

classification layer. [25] discussed the automated construction of lexicons containing abusive terms. They sampled 500 negative nouns, verbs and adjectives from the Subjectivity Lexicon by [26] and added 150 slangs like *ni**er*, *cunt*, *slut* etc. The word was classified into abusive only if 4 of 5 annotators voted it as abusive. Negative expressions from Wiktionary is utilized to expand the lexicon to 2989 abusive terms. Their proposed classifier is SVM trained on features derived from their expanded lexicons.

Based on the observation that hate speech also displays a high degree of negative polarity, several lexicons have been used to capture sentiment information as a feature. [27] considered *SentiWordNet*, *Affinn*, *Bing Liu*, *General Inquirer*, *Subjectivity clues* and *NRC* to explore the relationship between sentiment and toxicity in social media messages from 3 domains, namely Reddit, Wikipedia talk labels[5] and Toxic comment classification.[6] The toxicity detector is a Bi-GRU layer with words represented by 300d FastText pre-trained word embeddings [9], characters represented by 60 dimensions one-hot vector and 3 sentiment values obtained from 3 best lexicons based on their study. These input values are then concatenated together into a vector of 363 dimensions. [28] used Linguistic Inquiry and Word Count (LIWC) to count the frequency of words, that are indicative of various psychological processes, such as social words, cognitive processes, and affect words. They also made use of *TAACO* (Tool for the Automatic Analysis of Cohesion), a linguistic tool for automated analysis of text cohesion that provides more than 150 indicators of text coherence linguistic complexity, text readability, and lexical category use [29]. To find the negativity present in a post, IBM Watson Natural language Understanding API is used to compute the degree of anger, disgust, joy, fear, and sadness. They built an SVM classifier with a feature set of unigrams, TAACO, LIWC, sentiment values, and context to obtain AUC ROC of 0.74. [2] used the Vader sentiment analyzer by [30] to calculate the sentiment score of any tweet.

2.3. Meta-information

Meta-information of any tweet can also act as a useful feature to detect hate speech. [31] relied on crowd-workers to label 1.5k users as normal, spammers, aggressive, or bullies, from a corpus of $\approx$ 10k tweets distilled from a large set of 1.6M tweets. They proposed random forest classifier by investigating 30 features from 3 types of attributes (user, text, network-based) characterizing such behavior and found that bullies are less popular (fewer followers/friends, lower hub, authority, eigenvector scores) and participate in few communities. [32] presented a reinforced Bi-LSTM leveraging inter-user and intra-user representations for hate speech detection using the dataset by [4]. The intra-user tweet representation is obtained by analyzing m tweets posted by the user. Semantically similar to the target tweet from the set of unlabeled tweets were collected by *Locality Sensitive Hashing* [33].

[5] https://figshare.com/articles/Wikipedia_Talk_Labels_Toxicity/4563973.
[6] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

The intra-user information helps to reduce false positives of a model, which further improves by integrating inter-user similarity learning. [34] presented an approach purely based on graph features fed into SVM to tackle the problem of automatically detecting online abuse. They distinguished between *local features*, which characterize individual vertices, and *global features* that which describes the whole graph at once. The local topological measures are computed for the vertex corresponding to the author of the targeted message. These measures include *Degree Centrality*, *Eigenvector Centrality*, *PageRank Centrality*, *Betweenness Centrality*, *Closeness Centrality*, *Eccentricity*, *Coreness Score Hub and Authority Scores*. From the graph G= (V, E) where V and E are set of vertices and set of edges, graph topological measures like *Density*, *Diameter*, *Clique Count*, *Degree Assortativity* etc. were computed.

2.4. Linguistic study

Although the research community has been investigating the ways to employ different features for representing the tweets and building complex model, [35] highlighted the need for linguistic features indicative of different types of hate speech (e.g. racism, sexism, offensive, direct attacks, etc.), and argued that the type of datasets and labeling criteria is more important than the model architecture. [36] delineate the challenges in this field ranging from clarity in subtasks, lack of proper definition, linguistic difficulties in identifying content i.e (a). humor, irony, and sarcasm (b). Spelling variations (c). Polysemy (d). Long-range dependencies and (e). Language change. They also critically addressed the ethical challenges and discussed the importance of including textual contents that contain abuse but is not abuse in nature for the training. [37] presented a concrete methodology for annotating large and accurate datasets, performed statistical analysis for label-merging or label-elimination. [38] analyzed the concept of hate speech discussed so far in computer science applications and provided clear definitions to help in building automated detection tools. They also highlighted the absence of studies on hate speech detection in other than the English language.

2.5. Cross-domain information

Many studies have also been done on the cross-domain training of the classifier to correlate between the different sub-classes of hate. [39] primarily focused on various facets of abusive language and demonstrated that the proposed GRUs can be easily applied to detect abusive behavior in other online domains by utilizing metadata features like tweet-based, user-based and network-based features along with pre-trained word embedding. [40] investigated the cross-domain performance on 9 datasets using linear SVM and found that in-domain data is very important for training the model. They also explored the Frustratingly easy domain adaptation (FEDA) framework [41] to check the domain adaptability performance. [42] recommended cross-domain classification as a solution to deal with data bias that often goes unnoticed in in-domain classification.

2.6. Dealing with biases

Hate speech is a difficult phenomenon to define and it is important that we should be cognizant of the biases entering into the classification model. [43] provided a methodology for modifying an embedding to remove gender stereotypes. They evaluated their debiasing algorithms to ensure that it preserves the desirable properties of the original embedding while reducing both direct and indirect gender biases. [44] measured racial bias in hate speech and abusive language datasets. They highlighted the need to develop detection systems sensitive to different social and cultural contexts. [15] provided an examination of the influence of annotator knowledge of hate speech on classification results obtained from training on expert and amateur annotations. The model performance was increased, when trained on expert annotator's tweets over amateur annotators tweets. They suggested to calculating the weighted-F1 score to penalize misclassification on minority classes. [45] measured gender biases on the models trained with different abusive language datasets. They experimented with 3 bias mitigation methods: (i) *debiased word embeddings*, (ii) *gender swap data augmentation*, and (iii) *fine-tuning with a large corpus*. These methods reduce gender bias by 90%–98%.

2.7. Multi-task learning

Multi-tasking learning aims at solving more than one problem simultaneously. The end-to-end deep multi-task learning has been recently employed in solving various problems of Natural Language Processing (NLP), such as sentiment and emotion analysis [46–48] and text mining [49] etc. [50] defined MTL as follows:

Definition 1 (*Multi-Task Learning*)**.** Given m learning tasks

$$\{T_i\}_{i=1}^{m} \tag{1}$$

where all the tasks or subset of them are related, multi-task learning aims to help improve the learning of a model for classification task T_i by using the knowledge in some or all of the m tasks. [51] developed two forms of MTL, namely Symmetric multi-task learning (SMTL) and Asymmetric multi-task learning (AMTL). The former is joint learning of multiple classification tasks, which may differ in data distribution due to temporal, geographical, or other variations, and the latter refers to the transfer of learned features to a new task for the purpose of improving the new task's learning performance. [52] discussed the two most commonly used ways to perform multi-task in deep neural networks.

1. Hard Parameter Sharing: Sharing the hidden layers between all tasks with several task-specific output layers.
2. Soft Parameter Sharing: Each task has its own specific layers with some sharable part.

In this paper, we propose a deep multi-task learning framework to leverage the useful information of multiple related tasks. To deal with the data scarcity problem we utilize a multi-task learning approach that enables the model by sharing representations between the related tasks and generalize better by achieving better performance for the individual tasks. The proposed model is evaluated on five different datasets related to hate speech classification, racism detection, and offensive language detection. Detailed empirical evaluation shows that the proposed multi-task learning framework achieves statistically performance improvement over the single-task setting. Comparative evaluation results also reveal that the proposed approach outperforms the state-of-the-art systems over macro-F1 scores in the range of 10%-27%.

3. Methodology

3.1. Pre-processing

Social media posts contain a lot of noisy texts which are not considered as useful features for the classification. We perform the following steps to remove the noise, and make it ready for machine learning experiments:

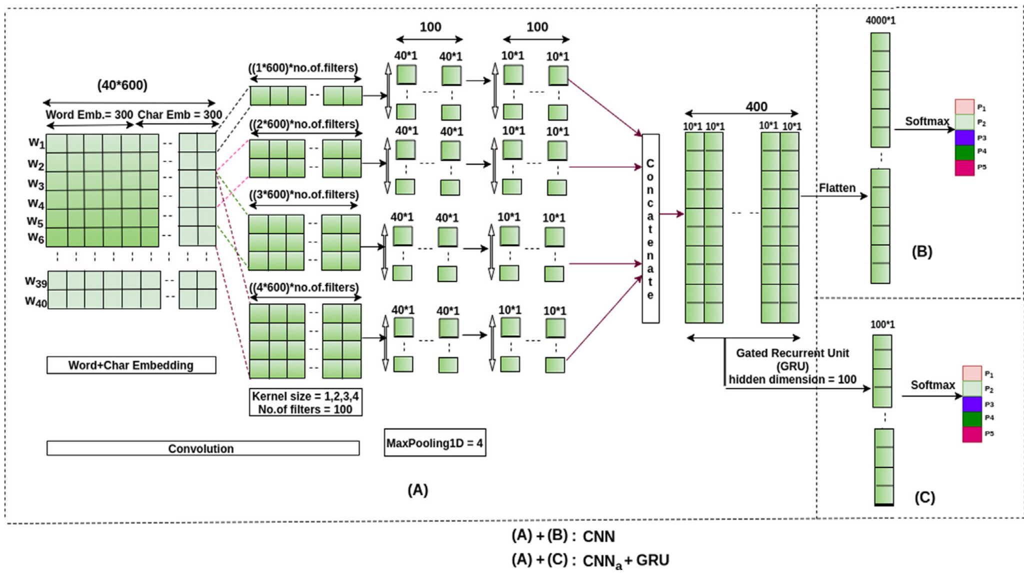

Fig. 1. Architecture of CNN and CNN_a+GRU.

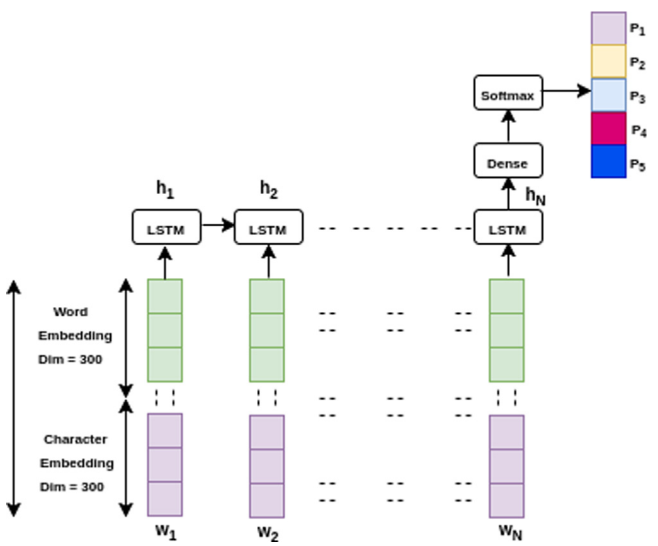

Fig. 2. Architecture of LSTM.

1. All the characters like |:,? were removed along with the numbers and URLs.
2. Words are reduced to lower case so that words such as "HAPPY", "happy" and "Happy" will have the same syntax and will utilize the same pre-trained embedding values.
3. Word segmentation is being done using the Python based *word segment*[7] to preserve the important features present in hashtag mentions. Some of the examples are: , #killerblondes → *killer blondes* , #suicideblondes → *suicide blondes*, #Iamcharliehebdo→ *I am charlie hebdo*, #whitegenocides →white genocides, #antiwhites → *anti whites*, #refugeeswelcome → *refugees welcome*.
4. All the emoticons were categorized into 5 categories, namely *love*, *sad*, *happy*, *shocking* and *anger*. The unicode character of emoticon in text is substituted with one category.
5. All the @ (ex.@abc) mentions were replaced with the common token, i.e *user*.
6. The stop words were not removed due to the risk of losing some useful information, and this was also empirically found to be of little or no impact on the classification performance after removing them.
7. The maximum sequence length is set to 40. Post padding is done if any sentence is less than 40 and pruning is performed from the last if the sentence is greater than 40.

3.2. Embedding layer

A sentence of length n can be represented as w_1, $w_2 \ldots w_n$ where each word can be represented as real valued vector.
Word Embedding (w_e): Two model architectures are generally used to compute the real-valued vector from the large data.
(i) **Continuous Bag of Words (CBOW) model**: This model predicts the current word from a window of surrounding context words.
(ii) **Continuous Skip-gram model**: The current word is used to predict the surrounding context words.

We use the *word2ec* model, pre-trained on 100 billion words and produce a 300-dimensional representation of each word, capturing the semantic and syntactic relationship between the words using skip-gram model [13].
Character Embedding (c_e): The presence of Out-of-Vocabulary (OOV) word is a serious problem in a social media text. Users in social media, to evade automatic checking, often perform intentional obfuscation of words by using short words, abbreviations, and misspelled words. Representation of such words in the pre-trained word embedding model is not found, thereby losing morphological information. We leverage the skip-gram model by [9], which represents each word as a bag of character n-grams. A vector value is associated with each character, the sum of these vector values represent the embedding for words. The dimension for character embedding is 300.

The final word embedding x_e for word $x \in X$ is represented by the following process:

$$x_e = w_e \oplus c_e \tag{2}$$

[7] http://www.grantjenks.com/docs/wordsegment/.

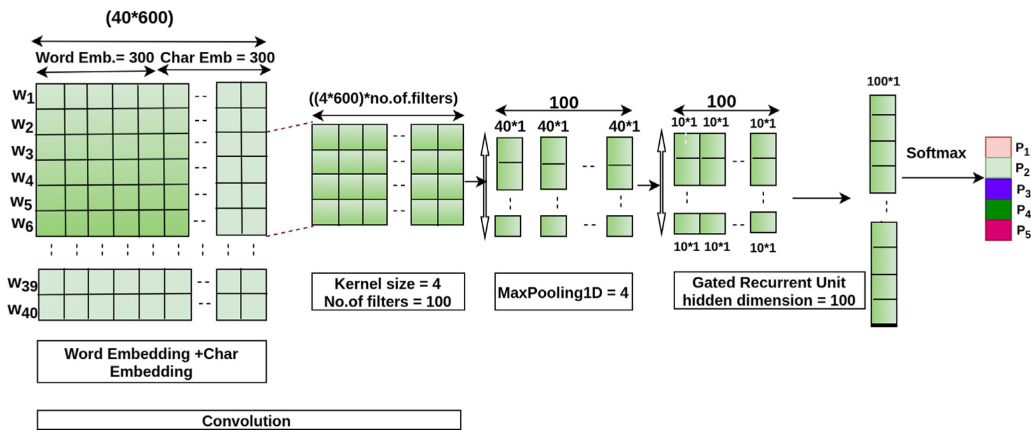

Fig. 3. Architecture of CNN+GRU.

where w_e is the word embedding, c_e is the character embedding, ($\oplus$) denotes the concatenation operation and X is the number of unique tokens. The resulting dimension of x_e is 600.

3.3. Single task learning (STL)

Most of the existing research on hate speech detection has focused on solving one classification task at a time. This is generally achieved by learning task-specific features from one dataset at a time. The shortcoming of the single data set is their small training samples and is related to a particular domain. In STL, a dataset D is used to train the neural network to perform a classification task T by mapping input sequences x_i to any predefined label y_i in a supervised manner. Each sentence x_i passes through a set of neural layers and final representation is passed through softmax to predict the probability distribution among C number of classes.

$$\hat{y} = softmax(Wh_T + b) \tag{3}$$

$$softmax_i = \frac{\exp(y_i)}{\sum_{j=1}^{C} \exp(y_j)} \tag{4}$$

where $\hat{y}$ is the prediction probability, W is the final optimal weight of the fully connected network after training, h_T is a hidden state and b is a bias term.

Given a corpus D with N training samples (x_i, y_i), the parameters of the network are trained to minimize the cross-entropy of the predicted and true distributions.

$$L(\hat{y}, y) = -\sum_{j=1}^{C}\sum_{i=1}^{N} y_i^j log(\hat{y}_i^j) \tag{5}$$

where y_i^j is the ground-truth label; $\hat{y}_i^j$ is the predicted label, and C is the class number.

3.4. Proposed models

We develop four deep neural networks, to be trained in single-task and multi-task paradigm. These are Convolution Neural Network (*CNN*) in Fig. 1, Long Short Term Memory (*LSTM*) in Fig. 2, Stacking of CNN and Gated Recurrent Unit (GRU) (*CNN+GRU*) in Fig. 3 and CNN_a*+GRU* by modifying *CNN+GRU*, and shown in Fig. 1.

1. **CNN based model**: For this, Convolution Neural network (CNN) for text classification by [53] was employed. CNN is a multi-layer trainable architecture, generally consisting of an input layer, embedding layer, convolution layer, pooling and fully connected layer. A sentence of length n (padded when required) can be represented as

$$X_{1:n} = x_1 \oplus x_2 \oplus x_3 \ldots x_n \tag{6}$$

where ($\oplus$) is the concatenation operator.
Input Layer: In this layer, all the words in the sequence are converted to their unique index u_i. The padding with 0 is performed to have an equal length for all the tweets.
Embedding Layer: Each unique index u_i of word w_i in the sentence is replaced by the embedding values E(u_i), a real-valued vector of n dimensions in the embedding matrix E.
Convolution Layer: It is one of the major components of the CNNs. It consists of learnable filter $w \in R^{hk}$, where h is number of words and k is the dimensions. This filter is used to extract features by convolving on h words at a time and performs an element-wise dot product to get a feature f_i. A feature f_i is generated from a window of words $X_{i:i+h-1}$ by

$$f_i = f(W * X_{i:i+h-1} + b) \tag{7}$$

Here, b ∈ R is a bias term, added as a parameter in neural network to fit best for any given data. f is an activation function introduced into the model to make it more powerful by adding ability into it to represent non-linear complex functional mapping between inputs and outputs. This filter is applied to each possible window of words in the sentence $\{X_{1:h}, X_{2:h+1}, \ldots X_{n-h+1:n}\}$ to produce a feature map F.

$$F = [f_1, f_2 \ldots . f_{n-h+1}] \tag{8}$$

N number of filters can be utilized to obtain N different feature maps.
Pooling Layer: It reduces the spatial size of the vector representations helping in coping up with overfitting. The most common is max over time pooling operation by [54] which takes the local maximum value depending on the pool size to capture the important features from the feature map.
Fully Connected layer: The final features obtained is then passed through the fully connected layer followed by the *softmax* layer that calculates the probability distribution over labels.

2. **LSTM based model** : RNN is very suitable for sequence learning, but as it suffers from vanishing gradient and exploding gradient it does not perform well for the long-range dependency problem. [55] introduced LSTM that is capable of learning long-range dependencies. It has internal mechanism called gates that regulate the flow of information. In LSTM there are 3 gates (i). Input Gate(i_t), (ii). Forget Gate(f_t), and (iii) Output Gate(o_t). Gates

Table 3
LSTM symbols.

i_t	Input gate
f_t	Forget gate
o_t	Output gate
b_i	Bias for the input gate neurons
b_f	Bias for the forget gate neurons
b_0	Bias for the output gate neurons
W_i	Weight for the input gate neurons
W_f	Weight for the forget gate neurons
W_o	Weight for the output gate neurons
σ	Sigmoid function with output in [0,1]
tanh	Hyperbolic tangent function with output in [−1,1]
$\odot$	Element wise multiplication
C_t	Current memory cell
W_t	Weight for the current memory cell
$\hat{C}_t$	Represents candidate for cell state at timestamp(t)
h_t	Current hidden state
h_{t-1}	Output of previous LSTM block timestep($t-1$)
C_{t-1}	Current memory cell at timestep($t-1$)
x_t	Input at current timestep t

are sigmoid function that values between 0 or 1. The equation for LSTM gates can be defined as follows.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{9}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{10}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{11}$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{12}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \tag{13}$$

$$h_t = \tanh(C_t) * o_t \tag{14}$$

$w_1, w_2, w_3 \ldots\ldots w_n$ represents the embedding values of a word in the sentence S whose length is N. $\{h_1, h_2, h_3 \ldots\ldots h_N\}$ represents the hidden vector for all the w_i where i $\in$ [1,n].

The last hidden vector h_N is considered as the representation of the S and fed h_N into a softmax layer after linearizing it into a vector whose length is equal to the number of class labels. Table 3 describes the various components of LSTM.

3. **CNN+GRU based model**: This is inspired by the architecture of [12] to automatically classify hate speech. We modified it to represent the word as the concatenation of its character embedding(c_e) and word embedding (w_e). It has an input layer followed by the drop out layer of 0.2. The output is then passed through a 1D convolution layer with 100 filters with a window size of 4 and ReLU as the activation function. The convolved input is down-sampled by a 1D max pooling layer with a pool size of 4. These extracted features are then fed into the GRU layer. Finally, a Softmax layer takes this vector as input to predict the probability distribution over all the possible classes. Fig. 3 explains the architecture.

4. CNN_a**+GRU based model**: This model utilizes the window sizes of 1,2,3 and 4 with 100 filters keeping the rest of the hyper-parameters as same in Model 3. The architecture of Model 4 is in Fig. 1.

3.4.1. Weight learning

Step 1: Initialization: Model initialization takes place with setting up of necessary hyperparameters in the model architecture.

Step 2: Forward propagation: The input sequence is passed through various layers of the neural network to compute the predicted class label.

Step 3: Loss function: It is defined as the performance of the Neural Network (NN) on how well it manages to reach the actual output. We utilize *categorical cross entropy* as the loss function for classification where C is the total number of classes, y_i^j is the ground truth label and $\hat{y}_i^j$ is the predicted label. The total loss can be defined as:

$$L(\hat{y}, y) = -\sum_{i=1}^{C}\sum_{j=1}^{N} y_i^j \log \hat{y}_i^j \tag{15}$$

Step 4: Optimization: In the next step, the model is optimized to minimize the loss function and find the weight W that minimizes the total loss. The weight update is done by optimizers like Adam, Adagrad, RMS Prop, etc.

Step 5: BackPropagation: The backpropagation algorithm finds the minimum value of error in loss function to find the optimal weight by delta rule or gradient descent. It checks whether to increase or decrease the weight values and updates the weight until error minimizes. The Derivative of the function is checked using two methods to get the optimum value.

1. If the derivative is positive, the error will get increased on increasing the weights. It means weight should be decreased.
2. If the derivative is negative, the error will decrease on increasing the weights. It means weight should be increased.

After each iteration, the gradient descent updates the weights towards lowering the global loss function. The weight update by *delta rule* is given as in Eq. (16) where *learning rate* is a constant value with 0.001. The learning rate is introduced to have a smooth update.

$$New\_weight = Old\_weight - Derivativerate * Learningrate \tag{16}$$

The number of iterations to achieve optimal weight depends on learning rate, meta parameters like number of layers, activation function, quality of training data, etc.

3.5. Shared-private multi task learning (SP-MTL)

The goal of multi-task learning is to utilize the correlation among the related tasks to improve the classification by learning data in parallel. In this paper, the Shared-Private MTL (SP-MTL) model by [56] is leveraged that introduces two feature spaces for each task: one is to store task-dependent features, the other is used to store task-invariant features. The training of SP-MTL involves three steps.

Step 1: Training of Shared Network (SN): The SN consists of 4 components: Shared Embedding Layer (SEL), Shared Neural Network (SNN), Shared Dense Layer (SDL) and Softmax layer. This network is pre-trained by taking equal samples from each of the 5 datasets and training it in batch-wise manner. The Shared Embedding Layer (SEL) consists of the unique tokens from 5 datasets. All the different class of dataset D_i is merged to represent class c_i where in this experiment $i \in [1, 5]$. Algorithm 1 explains the joint training of the data in Shared network. The parameters of the SN are trained to minimize the *categorical cross entropy* of the predicted and true distribution on all the tasks. The loss L_{Task} can be defined as:

$$L_{Task} = \sum_{k=1}^{K} \alpha_k \cdot L(\hat{y}^k, y^k) \tag{17}$$

where α_k is the class weight i.e 1 in this experiment and L($\hat{y}$,y) is defined in Eq. (15).

Step 2: The trained Shared network (SN) is sliced off to extract the weight matrix of the first two layers: SEL and SNN, denoted in red color in Fig. 4[8] The parameters of the transferred layers to the new network are kept frozen.

[8] The figure is best viewed in color.

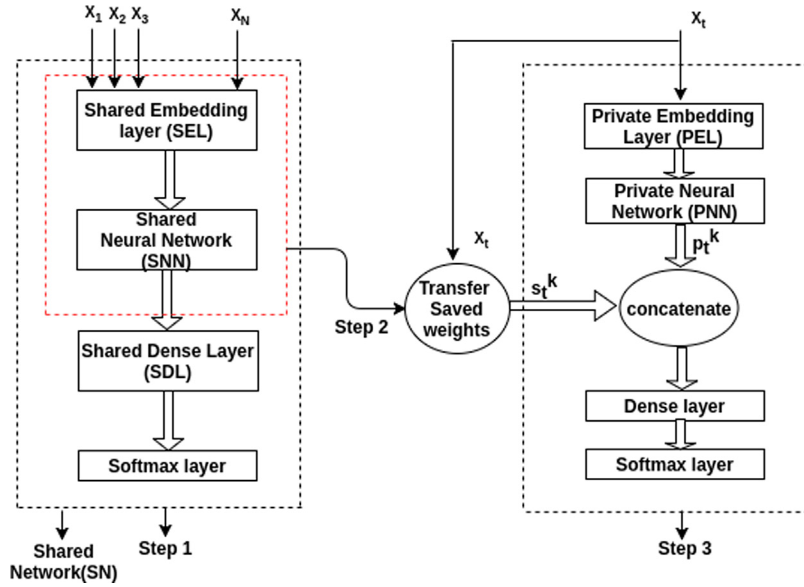

Fig. 4. Architecture of Shared-Private Multi Task Learning (SP-MTL).

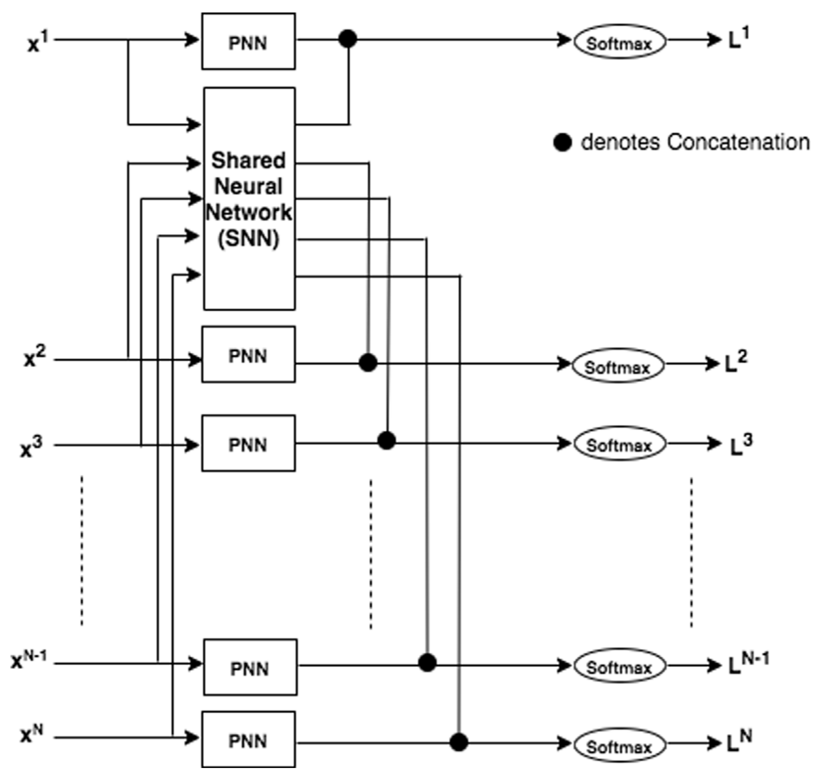

Fig. 5. Block diagram of shared-private multi-task learning (SP-MTL).

Step 3: The input sentence x_t is also passed through Private Embedding Layer (PEL) and Private Neural Network (PNN) to obtain p_t^k as private features. A sentence x_t of task k has shared representation s_t^k. s_t^k and p_t^k can be defined as follows.

$$s_t^k = NN(x_t, \theta_s) \tag{18}$$

$$p_t^k = NN(x_t, \theta_p) \tag{19}$$

where NN ∈ {CNN, LSTM, CNN+GRU, CNN_a+GRU}, θ_p and θ_s are the parameters of shared and private layers.

The shared features s_t^k and private features p_t^k for all the N tasks are concatenated to construct the architecture of Fig. 4. The vector is then passed to the dense layers and softmax layer of each classification task. Fig. 5 represents the block diagram of SP-MTL. We experimented with 4 types of (SP-MTL) frameworks, leading to 26 different combinations from these 5 datasets. The value of $N \in \{2, 3, 4, 5\}$ in Step 1, where as Step 2 and Step 3 are

Algorithm 1 Training of Shared Network(SN)

NN ∈ {CNN, LSTM, CNN+GRU, CNN_a+GRU}
Input: $D_1, D_2, D_3 \ldots D_{N-1}, D_N$ with x_j^i as training instance with class label $c_1, c_2, c_3 \ldots c_{N-1}, c_N$.
Batch_size = m, *Epochs* = n
Steps in one epoch = {maximum {len(D_1),len(D_2).....len(D_N)}/m } = maximum/m = Total
for *n* Epochs **do**
  for *Total* steps **do**
    sample *m* sentences (x_j^1,c_1) **from** D_1
    Update the Loss L_{Task}
    sample *m* sentences (x_j^2,c_2) **from** D_2
    Update the Loss L_{Task}
    sample *m* sentences (x_j^3,c_3) **from** D_3
    Update the Loss L_{Task}
....
....
....
    sample *m* sentences (x_j^{N-1},c_{N-1}) **from** D_{N-1}
    Update the Loss L_{Task}
    sample *m* sentences (x_j^N,c_N) **from** D_N
    Update the Loss L_{Task}
  end for
end for

same for all the 4 combinations. All the 4 combinations of SP-MTL are explained as following:

1. **Binary Shared-Private MTL:** Two input sequence x_a and x_b from D_i and D_j, where i ≠ j and i, j ∈ 1,2,3,4,5, are used to train the shared network with LSTM as SNN. Both the sequence x_a and x_b will utilize LSTM as their PNN in Fig. 5.

2. **Ternary Shared-Private MTL:** Three input sequences, *viz.* x_a, x_b and x_c from D_i, D_j and D_k, where i ≠ j ≠ k and i, j, k ∈ 1,2,3,4,5, are used to train the shared network with LSTM as SNN. x_a, x_b and x_c utilize LSTM as PNN in Fig. 5.

3. **Quaternary Shared-Private MTL:** Four input sequences, *viz.* x_a, x_b, x_c and x_d from D_i, D_j, D_k and D_l, where i ≠ j ≠ k ≠ l and i, j, k, l ∈ 1,2,3,4,5, are used to train the shared network with LSTM as SNN; and LSTM as PNN is used by all the four sequences: x_a, x_b, x_c and x_d in Fig. 5.

4. **Quinary Shared-Private MTL:** Five input sequences, *viz.* x_a, x_b, x_c x_d and x_e from D_i, D_j, D_k, D_l and D_m, where i ≠ j ≠ k ≠ l ≠ m and i, j, k, l and m ∈ 1,2,3,4,5, are used to train the shared network with SNN ∈ {CNN, LSTM, CNN+GRU, GNN_a+GRU } and PNN ∈ {CNN, LSTM, CNN+GRU, CNN_a+GRU } in Fig. 5, where *SNN* = *PNN*.

4. Dataset and terminologies

We evaluate our proposed multi-task model on 5 benchmark datasets. All the 5 datasets and terminology of each subtype of hate mentioned in the datasets are explained. Table 4 shows the statistics about the datasets.

D1: [2] {**Hate & Offensive**} collected seed slur terms to build lexicons from *Hatebase.org*. Using the Twitter API they searched for tweets containing terms from the lexicon and collected 85 million tweets from the account of 33,458 Twitter users. Then they took a sub-sample of ≈ 25K tweets from this corpus to annotate each tweet into one of the 3 classes. The corpus is annotated by a minimum of 3 annotators and a maximum of 9 annotators were involved for the annotation of some tweets. The Majority voting is followed to break the tie to decide the class label.

Table 4
Statistics of datasets used in the experiment.

Datasets	labels and count	#Posts	#Tokens
D1	Hate:1430 Offensive:19,190 Neutral:4163	24,783	16,362
D2	Racism:1923 Sexism:2871 Neutral:10,682	15,476	12,544
D3	OAG:3419 CAG:5297 NAG:6285	15,001	17,710
D4	Offensive:4400 Non-Offensive:8840	13,240	19,961
D5	Harassment:5285 Neutral:15,075	20,360	25,949

D2: [4] {**Racist & Sexist**} collected 136,052 tweets over 2 months and made public the annotated corpus of 16,924 tweets into 3 classes with their Tweet IDs. We were able to collect 15,476 tweets as many of the tweets were found to be deleted.[9] They utilized the criteria in *Critical Race Theory* to collect their corpus.

D3: [57] {**Aggression**} discusses different aggression and their types. The data for the corpus was crawled from Facebook Pages and Twitter. For Facebook, more than 40 popular pages of discussion among the Indians were recognized and crawled to collect the data. These pages include the news websites like NDTV News, ABP News, pages of political parties like INC, BJP as well as pages of student union like SFI, JNUSA. For Twitter, the data was collected using popular hashtags such as #beef ban, #election results, etc.

D4: [58] {**Offensive**} in OLID (Offensive language Identification Dataset) provides 3-layer hierarchy annotation of tweets. The first layer classifies the tweet into *Offensive* and *Non-Offensive*. The second layer is to classify the offensive posts into *targeted* and *untargeted*, and the third layer annotation is to classify targeted offensive into *individual*, *group* and *others*. They retrieved the examples in OLID from twitter using its API by searching for keywords and constructions that are often included in offensive messages, such as *she is *****, *you are *****, *antifa*, *MAGA*, *liberals*. In this paper, layer 1 annotation (*Offensive* and *Non-Offensive*) of this data set is used.

D5: [59] {**Harassment**}[10] This comprises of the tweets that use violence, including sexually violent language, degrading racist terms, vulgarity, threatening language, etc. The search terms like *#white genocide*, *#f**kni**ers*[11], *the jews*, *f**king faggot* etc. were utilized for collecting the corpus. They also reported statistics that 60% of users have witnessed someone being called by offensive names and 25% have been physically threatened.

Apart from the datasets used for this experiment, there are some other datasets from similar domains that have been created for building hate speech detection. Below, we discuss some of these data:

[18]: The dataset consists of ≈ 10K sentences labeled as *hate* or *no-hate*. The content was extracted from Stormfront using web-scraping techniques between 2002 and 2017. The most hateful words were *ape*, *scum*, *filthy*, *homosexuals*, *filth*, *monkey*, *libtard*, *coon*, *niglet* etc.

[60]: This dataset comprises of 56,280 comments containing *hate* speech comments and 895,456 *clean* comments generated by 209,776 anonymized users on Yahoo Finance Website.

[9] Due to deletion or suspension of account.
[10] The author kindly agreed to provide the data.
[11] Where present, the "*" has been inserted by us and was not part of the original text.

[12]: The authors used Twitter Streaming API to collect tweets containing the words frequent for hate speech: *kill*, *die*, *attack*, *terrorist*, *islam*, *immigrant*, *refugee*, *asylum*. They also retrieved tweets using hashtags like *#refugeesnotwelcome*, *#banislam*, *#norefugees* etc. The final dataset consists of 2435 tweets classified into *hate* and *non-hate*.

[37]: In this paper, the authors collected a random set of tweets utilizing Twitter Stream API during the period from 30th March 2017–9th April 2017, consisting of 32 million tweets in total. Finally, annotated dataset of ≈ 100k tweets labeled into *normal*, *spam*, *abusive* and *hateful* were released with Tweet ID. They provided a detailed methodology for annotating a large scale dataset.

[21]: The tweets have been collected from July 2018 to September 2018. The keywords that occur more frequently in the collected tweets are: *migrant*, *refugee*, *#buildthatwall*, *bi**h*, *h*e*, *women* etc. The final data consists of 13,000 tweets, out of which 5470 were tagged as *hate* and 7430 were tagged as *non-hate*.

4.1. Terminologies

Hate: [3] defined hate speech as any communication that disparages a person or a group based on some characteristics, such as race, color, ethnicity, gender, sexual orientation, nationality, religion or other characteristics.

Offensive: A type that describes the use of derogatory, hurtful and obscene comments made by someone to provoke reactions such as *anger*, *fear*, *disgust*, *outrage*, *hostility*, etc.

Racism: [61] defines *racism* as an ideology of racial domination in which cultural or biological superiority of one or more groups is justified for the inferior treatment of other racial groups. The united nations do not discuss racism, however, it does define *racial discrimination* in 1965 *International Convention on the elimination of all forms of racial discrimination.*

Sexism: The theory of sexism came into existence during the 1960 women liberation movement that asserted sexism as discrimination based on *sex* and *gender*. [62] proposed two subcomponents of sexism, i.e *hostile* and *benevolent*. The specified hostile versions are sexuality as combat, competing for gender roles, and male dominance in society. The benevolent versions are romantic intimacy, complementary gender roles, and women as cooperative subordinates. They observed male dominance leading to hostile behavior including violence, assault, and murder, whereas dependence upon the women fostered benevolent behavior. In the 1960s, the concept of *reverse sexism* documented referring to sexism against men and boys. [63] highlighted the theory of *second sexism*. However, the recent studies of data highlighted in [4] on 16K tweets showed that gender distributions to use hate speech are more skewed towards men with 50.08% users posting negative words as compared to 2.36% women and 47.64% of tweets were from the unidentified gender.

Aggression [64] defined aggression as any form of behavior directed towards another living being who is motivated to avoid such behavior. It can be *overt aggressive* or *covert aggressive* as described in the dataset by [57].

Overtly aggressive(OAG): Any speech/text in which aggression is overtly expressed either through the use of specific lexical items or lexical features. It uses direct verbal attacks like abusing someone, calling names in a derogatory manner pointed towards an individual or a group.

Covertly aggressive (CAG): It is an indirect attack against the victim and often packaged as insincere polite expressions. In general, these cases include metaphorical reference, satire, rhetorical questions etc.

Further, overtly aggression and covertly aggression can be classified into four types based on the target of aggression. (i). Physical Threat, (ii). Sexual Threat/Aggression, (iii). Identity Threat/Aggression, (iv). Non-Threatening Aggression.

Harassment [65] discussed the definition of *harassment* as a type of abuse in which user constructs the identity of sincerely wishing to be part of the group in question, including professing, or conveying pseudo sincere intentions, but whose real intention(s) is/are to cause disruption and/or to trigger or exacerbate conflict for the purpose of their own amusements.

The overlapping nature of the problems has been one of the motivations to leverage information from a variety of sources through deep multi-task learning. [66] highlighted the important distinctions between the diverse sub-tasks. They argued that the differences between the subtasks within abusive languages can be reduced to two primary factors:
(i). Is the language directed towards a specific individual or entity or is it directed towards a generalized group?.
(ii). Is the abusive content explicit or implicit?.

Tables 5–7 contain examples of different subtypes of hate. These instances show the severity of languages used in various sub-types of hate and give an idea to know which contents are of utmost priority to deal with. Each instance of hate can be grouped into two types, i.e *Explicit* and *Implicit*. The *explicit* tweets directly express hatred towards a particular target, whereas in the *implicit* tweets the hatred of target must be derived from the context. Following the approach proposed in [66] all the instances in Tables 5–7 are classified into four categories. These are *(i). Explicit attack towards individual* **(EI)**, *(ii) Explicit attack towards group* **(EG)**, *(iii). Implicit attack towards individual* **(II)**, and *(iv). Implicit attack towards groups* **(IG)**. The sentence belonging to *neutral* class are reported in two sets. One set contains sentences that are *Neutral* in nature and are not categorized into the above mentioned four types. The second set classifies *Neutral* class into EI, EG, II, and IG, as these contain abusive terms and sarcastic comments, but due to the combination of contextual information and protection for freedom of speech, these are tagged as non-hateful (c.f. Table 6).

5. Experimental setup and evaluation metrics

In this section, we present the experimental setups and the evaluation metrics used for the evaluation.

5.1. Experimental setup

All the deep learning models were implemented using Keras, a neural network package [67] with Tensorflow [68] as backend. We performed 5-fold cross-validation to use 80% for tuning the batch size and learning epochs and test the optimized model on 20% held-out data. We randomly split each training fold into 10% validation and 90% training. Categorical cross-entropy is used as a loss function and *Adam* optimizer is used for optimizing the network. All the models were trained for 3 epochs as this found to be (near)-optimal. The batch size of 30 is used for training *SN* Model and a batch size of 16 is utilized for extracting task-specific features using Private Neural Network in Fig. 5. The number of filters used in CNN is 100, and the kernel width ranges from 1 to 4. For the LSTM and GRU, the number of hidden nodes is set to be 100. The value for bias is randomly initialized to all zeros, *Relu* activation function is employed at the intermediate layer, and *Softmax* is utilized at the last dense layer.

Table 5
Examples of Sexism, Racism and Hate.

Class	Type	Sentence
Sexism	II	The menus look like they were made by a 5 year old little girl...in this case just the mental age of a 5 year old girl I guess #MKR
Sexism	II	Gay fiancé is not going to cope being away from the fresh meat #MKR
Sexism	EI	OH MY GOD. KAT IS SUCH A F*****G B***H. SO MUCH HATE #mkr
Sexism	EI	f**k off Kat! Douchebag. #MKR #bit**y
Sexism	EG	The girls are going through this is f***ed or are they s***ing Colin's c**k! #MKR
Sexism	EG	This is supposed to be a f*****g cooking show not a bunch of fa**ots walking stupidly down a runway #cringeworthy #mkr
Sexism	IG	I don't think men should be allowed to have opinions. @Rhace138 Call me sexist but I don't think women should be allowed to grow beards
Sexism	IG	RT @CarrotFuck Too many guys act like girls on twitter now a days, I'm not sexist but you don't have to act like a whiny b**ch
Racism	II	@UmarMal You sound like an Islamolunatic.
Racism	II	RT @davidjones720: OBAMA and his love for golf is greater that love for CHRISTIANS or Jews ! http://t.co/LludcU5jSP
Racism	EI	@nyazpolitics @urgedharry @greenlinerzjm Your religion is a murderous pile of filth. http://t.co/XW63vR3NMR
Racism	EI	@Isham_AlAssad @UN Nobody said that goat f***er. You made it up..
Racism	EI	@dankmtl Good night a**. Hope you get run over by a bus.
Racism	EG	@yuskan0723 The filthy Palestinians have attacked and murdered Jewish children in their schools in the past. So f**k Palestinians.
Racism	EG	@sajid_fairooz @IsraeliRegime Yes they are Jew hating barbarians.
Racism	IG	These girls are the equivalent of the irritating Asian girls a couple years ago. Well done, 7. MKR
Racism	IG	of course you were born in serbia...you're as f***ed as A Serbian Film #MKR
Hate	II	"@NoChillPaz: "At least I'm not a ni**er" http://t.co/RGJa7CfoiT" Lmfao
Hate	II	@MillerLite I don't follow you because everything about your company is white trash.
Hate	EI	"Who the f**k you callin jiggaboo, ni**a?!"
Hate	EI	@Brice2254 F**k you, fa**ot.
Hate	EG	California is full of white trash who moved from Oklahoma
Hate	EG	@RosieZaya1: Ur f****ng white trash
Hate	IG	Roman Catholics in the US are trash. They give Catholics in Europe a bad name.
Hate	IG	Those guys are the definition of white trash .

Table 6
Examples of Offensive, Harassment and Covertly Aggressive.

Class	Type	Sentence
Offensive	II	"@JasminePore: If you dressed up as a cat for Halloween you are basic." ...or a pussy
Offensive	II	"@chieflottahoes: In D.C., we call them rollers, in Chiraq we call them thots." All across America they have been known as a h*e.
Offensive	EI	"@YoungRicooo: Happy birthday b**ch n**ga @WexicanAfro702" Thanks pal
Offensive	EI	@SimplyHype: @kobebryant shut the f**k up, you f**king trash ni**a, work on another ring; http://t.co/6RleNKR7EE
Offensive	EG	RT @AthleteChamp: Destroying a team that talked trash before the game.
Offensive	EG	RT @BadAdviceDog: r u african because ur african b***h
Offensive	IG	;@RudePost: A new breed of white girl... https://t.co/0Zsu3zQiXD; she slammed that b**ch;
Offensive	IG	@Weakass_Drex: If Jennifer Hudson can lose weight anybody can you fat a*s h*es!!!
Harassment	II	@benshapiro You're on a roll today Ben! Obama can't even act like a good Christian let alone lecture the Jews!
Harassment	II	SOMEONE F****NG HIRE @lindsaylohan AS THEIR SPIN DOCTOR PLZ
Harassment	EI	RT @IrishDude10: Go back to prison you f***ing n**ger #Mayweather
Harassment	EI	@Bambi_cop Untag you f*****g ni**er
Harassment	EG	RT @TrapHouseMario: White people literally Be like " I'm not racist you f***ing Ni**er"
Harassment	EG	@GORBtheOTHER The Jews are the masterminds idiot
Harassment	IG	RT @janimine: Diversity is a code word for #WhiteGenocide https://t.co/A28ASFSGR4
Harassment	IG	@AmancioHitler I said "a glass of juice" not "gas the Jews"!!
CAG	II	Anna hazare is agent
CAG	II	Anirban Kundu So you are the pesticide
CAG	EI	Anna is a "Natak Raja" !!!
CAG	EI	So that Modi doesnt have to campaign separately... Feku and his never ending gimmicks
CAG	EG	Also try to stop violence from communists
CAG	EG	Judiciary system of India.........????????
CAG	IG	The Great Powers should avoid such dangerous inhuman dramas to happen in the society.. . it creates bad results on one's health
CAG	IG	Well first of all municipal corporation need to keep cows off the streets. Its cows owner's responsibility.

5.2. Evaluation metrics

Macro-F1 and Weighted-F1 have been used to report the evaluation results, and to be consistent with the previous state-of-the-arts. The Precision(P_i), Recall (R_i), F1-score (F_i), Macro-F1 and Weighted-F1 can be defined as:

$$P_i = \frac{TruePositive}{TruePositive + FalsePositive} \tag{20}$$

$$R_i = \frac{TruePositive}{TruePositive + FalseNegative} \tag{21}$$

$$F_i = 2 \cdot P_i \cdot R_i \tag{22}$$

$$Macro\_F1 = \frac{1}{C}\sum_{c=1}^{i=c} F_i \tag{23}$$

$$Weighted\_F1 = \frac{1}{N}\sum_{i=1}^{i=n} F_i \cdot N_i \tag{24}$$

$$N = \sum_{i=1}^{i=n} N_i \tag{25}$$

6. Results

In this section we report the evaluation results in details along with the necessary analysis.

Table 7
Examples of Overtly Aggressive and Neutral.

Class	Type	Sentence
OAG	II	Because you are dalaal
OAG	II	You protect cows more than the women and children in this country. Thats pathetic!!
OAG	EI	Fcuk off Shikha Sharma
OAG	EI	Digvijay singh should be sent to pakistan to permanently reside there & keep dialogue with them.
OAG	EG	I hate both loud speakers whether played by Hindu or Muslim.
OAG	EG	Dear Indian Express, I feel sorry to write that your journalists are out there to misinterpret events and present them in a distorted manner just to make some sensational headlines.
OAG	IG	Party of jokers.
OAG	IG	Communist dogs need strong handling
Neutral	II	Nikki...Kermit the frog called and he wants his voice back #MKR #MKR2015 #KillerBlondes @mykitchenrules
Neutral	II	Her laugh is horrendous #MKR
Neutral	EI	#MKR F*ck you Colin! That was total shite!
Neutral	EI	Kat is definitely an A GRADE B***H, but channel 7 will be loving it, she's a ratings puller. Everyone loves to hate her #mkr
Neutral	EG	@amherstuprising no f****ng whiteys allowed.
Neutral	EG	@ChinaWhite_ because feminists are f***ing loons tbh
Neutral	IG	'This is our #Israel, this is for the Jews. No #Palestinian should come to Israel.' http://t.co/Fk2x7QHfqT http://t.co/uMRapTVMeE
Neutral	IG	RT @TRAPFUHRER: I said a GLASS OF JUICE not GAS THE JEWS. You dumb a*s hell fam
Neutral	–	Karanjohar better look at it
Neutral	–	Hi Manisha. Doing excellent work keep it up Good luck
Neutral	–	We should learn synonym of all the words from this man ;)
Neutral	–	Love you haters
Neutral	–	She is not a neurosurgeon, false news being spread again and again by the mainstream media.
Neutral	–	What about Goa?
Neutral	–	Hope they do not tax this is as well..hope Govt waives the Tax
Neutral	–	Shame for all of us.
Neutral	–	Seven of eight of saarc countries is the part of great cpec project even whole globe wants to part of cpec' but a Gelious india did'nt to part of great progress of the region
Neutral	–	Nifty 5300 in 2010–11, 2016 we are at 8700. Don't you feel nifty racing towards 12000
Neutral	–	Politics should not come in front... and play with it
Neutral	–	Why differentiate between silver and bronze ....both r doing well for country

Table 8
Evaluation results on learning 2 classification tasks jointly ((*BinaryShared*) — *PrivateMTL*).

	D1		D2		Common tokens
D1–D2	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	87.09(+25)	95.08(+8)	90.92(+12.55)	93.25(+9)	7230
	D1		D3		Common tokens
D1–D3	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	88.03(+26)	0.9548(+8.69)	84.28(+28.4)	84.75(+**27.36**)	7545
	D1		D4		Common tokens
D1–D4	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	88.47(+**26.38**)	95.48(+**8.69**)	90.86(+18.31)	91.86(+15.82)	7715
	D1		D5		Common tokens
D1–D5	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	84.76(+22.67)	94.18(+7.39)	85.58(+27)	88.94(+17.32)	9613
	D2		D3		Common tokens
D2–D3	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	90.18(+11.81)	92.52(+8.65)	83.56(+27.68)	84.08(+26.69)	7325
	D2		D4		Common tokens
D2–D4	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	90.62(+12.25)	92.96(+9.09)	91.37(+**18.82**)	92.36(+**16.32**)	7231
	D2		D5		Common tokens
D2–D5	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	90.98(+**12.61**)	93.17(+**9.3**)	84.98(+26.44)	88.61(+16.99)	8469
	D3		D4		Common tokens
D3–D4	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	83.76(+**27.88**)	84.35(+26.96)	91.30(+18.75)	92.30(+16.26)	7750
	D3		D5		Common tokens
D3–D5	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	82.84(+26.96)	83.44(+26.05)	85.06(+26.52)	88.71(+17.09)	9376
	D4		D5		Common tokens
D4–D5	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	90.38(+17.83)	91.48(+15.44)	85.73(+**27.19**)	89.18(+**17.56**)	9153

Table 9
Evaluation results on learning 3 classification tasks jointly ($(TernaryShared) - PrivateMTL$).

D1–D2–D3	D1		D2		D3		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	88.48(+**26.39**)	95.37(+8.58)	91.96(+13.59)	94.04(+10.17)	83.21(+27.33)	83.88(+26.49)	5409
D1–D2–D4	D1		D2		D4		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	87.95(+25.86)	95.41(+8.62)	90.18(+12.32)	92.69(+8.82)	90.67(+18.22)	91.69(+15.65)	5516
D1–D2–D5	D1		D2		D5		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	88.21(+20.12)	95.39(+8.60)	92.12(+**13.75**)	94.10(+10.23)	85.42(+26.88)	88.83(+17.21)	6170
D1–D3–D4	D1		D3		D4		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	87.70(+25.61)	95.40(+8.61)	82.66(+26.78)	83.32(+25.93)	90.40(+17.85)	91.50(+15.46)	5537
D1–D3–D5	D1		D3		D5		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	88.67(+25.58)	95.67(+**8.88**)	82.27(+26.39)	82.86(+25.47)	86.03(+**27.49**)	89.32(+**17.70**)	6278
D1–D4–D5	D1		D4		D5		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	87.37(+25.28)	95.22(+8.43)	90.44(+17.89)	91.57(+15.53)	85.36(+26.80)	88.84(+17.22)	6448
D2–D4–D5	D2		D4		D5		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	90.60(+12.23)	94.08(+**21.53**)	91.08(+18.53)	92.06(+16.02)	85.19(26.65)	88.67(+17.05)	6195
D3–D4–D5	D3		D4		D5		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	83.86(+27.98)	84.34(+27.95)	91.22(+**18.67**)	92.24(+**16.20**)	85.79(+27.25)	89.16(+17.54)	6384
D2–D3–D4	D2		D3		D4		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	92(+13.63)	93.89(+10.02)	83.96(+28.08)	84.56(+**28.17**)	91.21(+18.66)	92.20(+16.16)	5625
D2–D3–D5	D2		D3		D5		Common tokens
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	
	90.94(+12.57)	93.05(+9.18)	84.04(+**28.16**)	84.42(+27.03)	85.38(+26.84)	88.84(+17.22)	6177

Table 10
Evaluation results on learning 4 classification tasks jointly ($(QuaternaryShared) - PrivateMTL$).

A[a]	D1		D2		D3		D4	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
	88.21(+26.12)	95.54(+8.75)	90.10(+11.73)	92.68(+8.81)	83.46(+27.58)	84.09(+26.70)	90.75(+18.20)	91.85(+15.81)
B[b]	D1		D2		D3		D5	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
	89.07(+26.98)	95.70(+8.91)	90.90(+12.53)	93.23(+9.36)	83.46(+27.58)	84.16(+26.77)	86.05(+**27.51**)	89.38(+**17.76**)
C[c]	D1		D2		D4		D5	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
	89.30(+**27.21**)	95.92(+**9.12**)	90.81(+12.44)	93.21(+9.34)	91.33(+**18.78**)	92.30(+**16.26**)	85.73(+27.19)	89.08(+17.46)
D[d]	D1		D3		D4		D5	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
	89.18(+27.09)	95.82(+9.03)	83.32(+27.44)	83.89(+26.50)	90.88(+18.33)	91.93(+15.89)	85.33(+26.79)	88.95(+17.33)
E[e]	D2		D3		D4		D5	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
	91.13(+**12.76**)	93.23(+**9.36**)	84.31(+**28.43**)	84.88(+**27.49**)	90.94(+18.39)	91.94(+15.90)	85.88(+27.34)	89.22(+17.60)

[a] D1–D2–D3–D4.
[b] D1–D2–D3–D5.
[c] D1–D2–D4–D5.
[d] D1–D3–D4–D5.
[e] D2–D3–D4–D5.

6.1. Single task learning (STL) vs shared-private multi task learning (SP-MTL)

All the four neural network models are trained in STL and SP-MTL manner on all the datasets, *viz.* D1, D2, D3, D4 and D5 to perform the classification task. Tables 8–10 denote the macro-F1 and weighted-F1 obtained by *Binary Shared-Private MTL*, *Ternary Shared-Private MTL* and *Quaternary Shared-Private MTL*. The macro-F1 and weighted-F1 are reported for each dataset in Tables 11, 13, 15, 17 and 19 respectively when all 5 datasets were trained jointly in SP-MTL fashion. The confusion matrices obtained using best neural network based models for all datasets are reported in Tables 12, 14, 16, 18 and 20 respectively.

The noteworthy performance improvement in terms of macro-F1 and weighted-F1 in Shared-Private MTL over STL are given in bracket in Tables 8–11, 13, 15, 17 and 19. The evaluation

Table 11
Evaluation results on D1.

Models	STL		SP-MTL	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
CNN	71.78	90.14	**89.16**(+**17.38**)	**95.65**(+5.51)
LSTM	62.09	86.79	88.40(+26.31)	95.65(+8.86)
CNN+GRU	**73.47**	**90.26**	88.70(+15.23)	95.58(+5.32)
CNN_a+GRU	71.98	90.01	87.41(+15.43)	95.01(+5.0)

Table 12
Confusion matrix of D1.

STL				SP-MTL			
Class	Hate	Offensive	Neutral	Class	Hate	Offensive	Neutral
Hate	429	869	132	Hate	1028	335	67
Offensive	376	18,330	484	Offensive	215	18,742	233
Neutral	46	348	3769	Neutral	28	186	3949

Table 13
Evaluation results on D2.

Models	STL		SP-MTL	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
CNN	77.43	83.27	**91.15**(+**13.72**)	93.31(+10.04)
LSTM	78.37	83.87	91.09(+12.72)	**93.35**(+**9.48**)
CNN+GRU	79.36	**84.43**	89.99(+10.63)	92.77(+8.34)
CNN_a+GRU	**79.55**	84.32	88.49(+8.94)	91.38(+7.06)

Table 14
Confusion matrix of D2.

STL				SP-MTL			
Class	Racism	Sexism	Neutral	Class	Racism	Sexism	Neutral
Racism	1544	8	371	Racism	1746	8	169
Sexism	10	1837	1024	Sexism	22	2446	403
Neutral	494	459	9729	Neutral	166	261	10,255

Table 15
Evaluation results on D3.

Models	STL		SP-MTL	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
CNN	55.04	57.13	**86.12**(+**31.08**)	**86.56**(+**29.43**)
LSTM	55.88	57.39	83.71(+27.83)	84.31(+26.92)
CNN+GRU	55.69	57.38	80.34(+24.65)	81.08(+23.70)
CNN_a+GRU	**56.03**	**58.04**	76.93(+20.90)	77.78(+19.74)

Table 16
Confusion matrix of D3.

STL				SP-MTL			
Class	OAG	CAG	NAG	Class	OAG	CAG	NAG
OAG	1310	1712	397	OAG	2846	367	206
CAG	688	3261	1348	CAG	316	4461	520
NAG	238	1896	4151	NAG	165	438	5682

Table 17
Evaluation results on D4.

Models	STL		SP-MTL	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
CNN	72.73	76.66	**92.41**(+**19.68**)	**93.31**(+**16.65**)
LSTM	72.55	76.04	90.80(+18.25)	91.83(+15.79)
CNN+GRU	73.0	76.98	80.34(+7.34)	81.08(+4.10)
CNN_a+GRU	**74.83**	**78**	87.01(+12.18)	88.54(+10.54)

results points to the fact that all the different sub-types of hate help each other to provide more evidences and contextual information, resulting in better classification performance. Experiments show that D_1 achieves the best performance when trained with $D_1D_2D_4D_5$. D_2 achieves the best results when trained with $D_1D_2D_3$ and $D_1D_2D_5$. D_3 reports to have achieved the best results when trained with $D_1D_2D_3D_4D_5$. D_4 shows best performance when trained with $D_1D_2D_3D_4D_5$. D_5 obtains the best performance when trained with $D_1D_2D_3D_4D_5$ and $D_1D_2D_3D_4D_5$.

6.2. Comparison to the state-of-the-art systems

Tables 21–25 report the comparison between State-of-the-art system and the proposed approach on the basis of macro-F1 and weighted-F1 for each task. The state-of-the-art systems for each data is explained as following:

Comparison to the state-of-the-art systems and proposed system for D1

(i) [2]: They represented text sequences with unigram, bigram, trigram weighted by their TF–IDF (term frequency–inverse document frequency). They also created PoS (Part of Speech) unigram, bigram, and trigram along with other handcrafted features in Logistic Regression (LR) with L2 regularizer to obtain weighted-F1 of 90%.
(ii) [39]: They utilized GloVe [10] word vectors and a set of three metadata features, namely tweet-based, user-based and network-based features to obtain weighted-F1 of 89%.
(iii) [69]: created stacked Bi-LSTM with contextual attention to obtain weighted-F1 of 91.10%.
(iv) [70]: proposed an ensemble learning of CNN, LSTM, Bi-LSTM, Bi-GRU, Bi-GRU-Attention and LR with char n-grams and word n-grams features fed into it to obtain macro-F1 of 79.30%.
(v) [16]: They utilized BERT (Bidirectional Encoder Representations from Transformers) [17], a recent transformer-based pre-trained contextualized embedding model extendable to a classification model with an additional output layer to obtain macro-F1 of 89.17%.
(vi) [71]: They proposed deep context-aware embedding that consists of two main modules: deep hybrid contextual word representation and BiLSTM with attention. This model obtained weighted-F1 of 92.30%.

Proposed Approach: The proposed approach obtains 89.30% macro-F1 and 95.92% as weighted-F1 when D1D2D4D5 were trained jointly in SP-MTL (Table 10)

Comparison to the state-of-the-art systems and proposed system for D2

(i) [4]: utilized gender and location of the user as the features and employed it in the logistic regression classifier with char n-grams to conclude that gender and location could not help to improve the performance of the F1 score to a significant level. The system obtained macro-F1 of 73.93%.
(ii) [72]: They employed an LSTM to task-tune GloVe initialized word-embedding followed by training gradient boosted decision tree (GBDT) to classify text based on the average of word embedding concatenated with char n-grams to obtain 79.31% macro-F1.
(iii) [73]: This model uses the 300 dimensional GloVe word embeddings. Each word embedding is then transformed by applying 300 dimensional 1-layer multi-layer perceptron (MLP) with *Relu* to create a transformed word embedding model (TWEM). It allows for better handling of infrequent and unknown words to obtain weighted-F1 of 86%.
(iv) [39]: They considered content-based features such as the number of hashtags, number of emoticons, sentiment scores, etc. User-based features such as the number of followers/friends, subscribed list, etc. and network-based features to obtain weighted-F1 of 87%.
(v) [74]: They proposed Hybrid CNN i.e a combination of word n-gram based CNN and character n-gram based CNN to get 83% weighted-F1.

Table 18
Confusion matrix of D4.

STL			SP-MTL		
Class	Offensive	Non-Offensive	Class	Offensive	Non-Offensive
Offensive	2703	1697	Offensive	3837	563
Non-Offensive	1165	7675	Non-Offensive	315	8525

Table 19
Evaluation results on D5.

Models	STL		SP-MTL	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
CNN	58.23	71.57	85.26(+27.03)	88.77(+17.20)
LSTM	58.54	71.62	**85.35(+26.81)**	**88.81(+17.19)**
CNN+GRU	**60.03**	**72.55**	82.73(+22.70)	90.85(+18.30)
CNN_a+GRU	58.17	71.58	77.04(+18.87)	82.61(+11.03)

Table 20
Confusion matrix of D5.

STL			SP-MTL		
Class	Harassment	Neutral	Class	Harassment	Neutral
Harassment	1206	4079	Harassment	4045	1240
Neutral	605	14,470	Neutral	1021	14,054

Table 21
Comparison to the state-of-the-art systems and proposed system for D1.

Authors	Macro-F1	Weighted-F1
[2]	–	90
[39]	–	89
[69]	–	91.10
[70]	79.3	
[16]	89.17	
[71]	–	92.3
Proposed approach	**89.30**	**95.92**

Table 22
Comparison to the state-of-the-art systems and proposed system for D2.

Authors	Macro-F1	Weighted-F1
[4]	73.93	
[72]	79.80	
[75]	79.24	84.14
[76]	80.49	
[73]	–	86
[39]	–	87
[74]	–	83
[69]	–	84.25
[71]	–	85.5
[8]	–	93
[77]	–	93.20
Proposed approach	**92.12**	**94.10**

(vi) [75]: This CNN based model with 2 convolution layers of size *64*3* and *64*4* with GloVE embeddings obtained 79.24% macro-F1 and 84.14% weighted-F1.
(vii) [76]: They used e**X**treme **G**radient **B**oosting (XGBoost) with simple character n-grams and word n-grams to obtain 80.49% macro-F1.
(viii) [69]: This system achieved 84.25% weighted-F1 using stacked Bi-LSTM with contextual attention.
(ix) [71]: The model proposed here have used deep contextual embedding with BiLSTM that can handle issues of polysemy semantics, syntax and OOV words to achieve 85.5% weighted-F1.
(x) [8]: The best method proposed here is *"LSTM+ Random Embedding +GBDT"*, where tweet embeddings were initialized to random vectors, LSTM was trained using back-propagation and then learn embeddings were used to train a GBDT classifier to get 93% weighted-F1.

Table 23
Comparison to the state-of-the-art systems and proposed system for D3.

Authors	Macro-F1	Weighted-F1
[78]	–	58.30
[78]	–	58.72
Proposed approach	**86.12**	**86.56**

Table 24
Comparison to the state-of-the-art systems and proposed system for D4.

Authors	Macro-F1	Weighted-F1
[80]	71.66	–
[80]	78.267	–
[81]	73.82	–
[81]	72.85	–
Proposed approach	**92.41**	**93.31**

(xi) [77]: The ensemble based LSTM classifier with randomly initialized word embedding with a vector size of 30 and user tendency towards posting racism, sexism and neutral tweets as features obtained 93.20% weighted-F1.

Proposed Approach: It obtains the macro-F1 of 92.12% and weighted-F1 of 94.10% when D1D2D5 were trained jointly in SP-MTL (Table 9)

Comparison to the state-of-the-art systems and proposed system for D3

[78]: The BiLSTM is trained on google pre-trained word2vec. This feature is then passed through the softmax layer to obtain a weighted-F1 of 58.30%.
[78]: This model is leveraging the concatenation of the features obtained from 2 neural networks. The BiLSTM is utilizing the google pre-trained embedding of dimension 300 whereas Character CNN by [79] is using character embedding obtained by a one-hot encoding approach. This system is obtaining weighted-F1 of 58.72%.

Proposed Approach: The proposed MTL approach shown improvement of up to 28% in weighted-F1 to score 86.56%.

Comparison to the state-of-the-art systems and proposed system for D4

(i) [80]: The LSTM based approach utilizes one-hot vector representation for the words, and obtained macro-F1 of 71.66%.
(ii) [80]: They proposed BERT based model that uses a multi-head transformer structure pre-trained on the huge corpus from different sources. It obtained 78.26% macro-F1.
(iii) [81]: The Bi-LSTM with Glove word embedding produced the macro-F1 of 73.82%.
(iv) [81]: They devised an architecture that combines both BiGRU ⊕ BiLSTM. The embedded words are processed in parallel through two branches of BiLSTM-CNN and BiGRU-CNN to get macro-F1 of 72.85%.

Proposed approach: Our proposed system utilized the multiple contextual information from multiple tasks to outperform BERT based classifier by getting 92.41% macro-F1 and 93.31% weighted-F1.

Comparison to the state-of-the-art systems and proposed system for D5

(i) [69]: The stacked BiLSTM with contextual information achieved 72.75% weighted-F1.

Table 25
Comparison to the state-of-the-art systems and proposed system for D5.

Authors	Macro-F1	Weighted-F1
[69]	–	70.57
[69]	–	72.75
[73]	–	71
[71]	–	73.6
[82]	70	–
Proposed approach	86.05	90.85

Table 26
Test data statistics.

Data	Class	Total
TRAC-FB	OAG:144 CAG:141 NAG:627	912
TRAC-SM	OAG:361 CAG:413 NAG:627	1257
OLID	Offensive:240 Non-Offensive:620	860

(ii) [69]: The stacked BiLSTM with self-attention achieved 70.57% weighted-F1.
(iii) [73]: The Transformed Word Embedding Model (TWEM) computed 71% weighted-F1.
(iv) [71]: They leveraged the power of sentiment analysis and used deep contextual embedding to get 73.6% weighted-F1.
(v) [82]: The LSTM based approach using Word2Vec, GloVe and Sentence specific word embeddings (SSWE) embedding obtained macro-F1 of 70% in all three cases.

Proposed Approach: It registered over 15% improvement in macro-F1 and weighted-F1 by obtaining 86.05% and 90.85% respectively.

6.3. Knowledge transfer

After the training of SN in Algorithm 1 (Step 1 in Section 3.5) with 88860 tweets, any input sequence on passing through it will generate a shared feature representation. To test the efficacy of the shared features the weight matrix of upper 2 layers in red block in Fig. 6[12] of Shared Network (SN) were transferred to a new network N_T. The parameters of transferred layers are kept frozen and parameters of the rest of the layers are randomly initialized. In Step 2, a new input sequence n_i is passed through the frozen transferred layers followed by dense layer to provide probability distribution over the classes.

We use 3 small test datasets shown in Table 26: TRAC-FB (facebook) [57], TRAC-SM (social media) [57] and OLID [83] for the evaluation. Tables 27 and 29 depict the results obtained by using all the four models as Shared Neural Network (SNN) framework as shown in Fig. 6. Tables 28 and 30 reports the confusion matrix for the TRAC test data and OLID data respectively.

TRAC-FB and TRAC-SM

In the following we are explaining in brief the state-of-the-results on these test sets with the proposed approach.
[84]: presented a model comprised of multi-dimension capsule network to generate the representation of sentences. This model obtained weighted F-score of 57.95% and 63.53% on Facebook and social media test set.
[85]: designed LSTM for facebook data to obtain 64.25% weighted-F1 while combination of CNN-LSTM is performing best for social media data by obtaining 59.20% weighted-F1.

[12] The figure is best viewed in color.

Table 27
Results on TRAC-FB and TRAC-SM.

Models	Facebook		Social media	
	Macro-F1	Weighted-F1	Macro-F1	Weighted-F1
CNN	57.13	72.81	80.25	80.91
LSTM	41.05	63.95	74.18	75.44
CNN+GRU	67.04	78.32	79.42	**86.52**
CNN_a+GRU	**69.53**	**80.68**	**82.81**	83.48
[84]	–	57.95	–	63.53
[85]	–	64.25	–	59.20
[86]	–	63.15	–	57.16

Table 28
Confusion matrix of *TRAC-FB* and *TRAC-SM*.

TRAC-FB				TRAC-SM			
Class	OAG	CAG	NAG	Class	OAG	CAG	NAG
OAG	66	15	63	OAG	291	61	9
CAG	15	70	56	CAG	71	300	42
NAG	8	2	617	NAG	6	16	461

Table 29
Results on OLID test data.

Models	OLID	
	Macro-F1	Weighted-F1
CNN	77.53	82.93
LSTM	72.63	79.38
CNN+GRU	84.29	87.67
CNN_a+GRU	**84.92**	**88.31**
[80]	82.9	86.24
[87]	81.40	85.33
[88]	80.80	84.59

Table 30
Confusion matrix of OLID.

Class	Offensive	Non-Offensive
Offensive	163	77
Non-Offensive	19	601

[86]: combined the Singular value decomposition (SVD) with TF–IDF and SVM to get the weighted-F1 of 63.15% and 57.16% on facebook and social media test data respectively.

• The proposed approach of transferring shared knowledge is showing improvement in up to 16% F-score.

OLID

The proposed approach of transferring weight to a new network outperformed the [80] BERT model by 2% in macro-F1 and weighted-F1. The system by [87] and [88] also used BERT based classification model.

7. Error analysis

Quantitative Analysis: The sentences in *Neutral* class play a very crucial role in determining the annotators' global knowledge about any specific topic and how much they can distinguish between free speech or any subtypes of harmful speech. In the SP-MTL for D1 in Table 12, only 28 neutral instances are misclassified to hate showing the model's ability to distinguish the hateful text. In the SP-MTL for D2 in Table 14, the false-positive of neutral class to racism and sexism reduced to 166 and 261 showing the efficacy of MTL. For D3, the accuracy of identifying NAG increased from 66% in STL to 90% in SP-MTL. In D4, the true positive rate of non-offensive class significantly improved by 8.2% in the SP-MTL setting. Surprisingly the false negatives increased for the non-harassment class in the SP-MTL paradigm. It can be concluded that SP-MTL outperforms STL in all the aspects. In Table 31, the

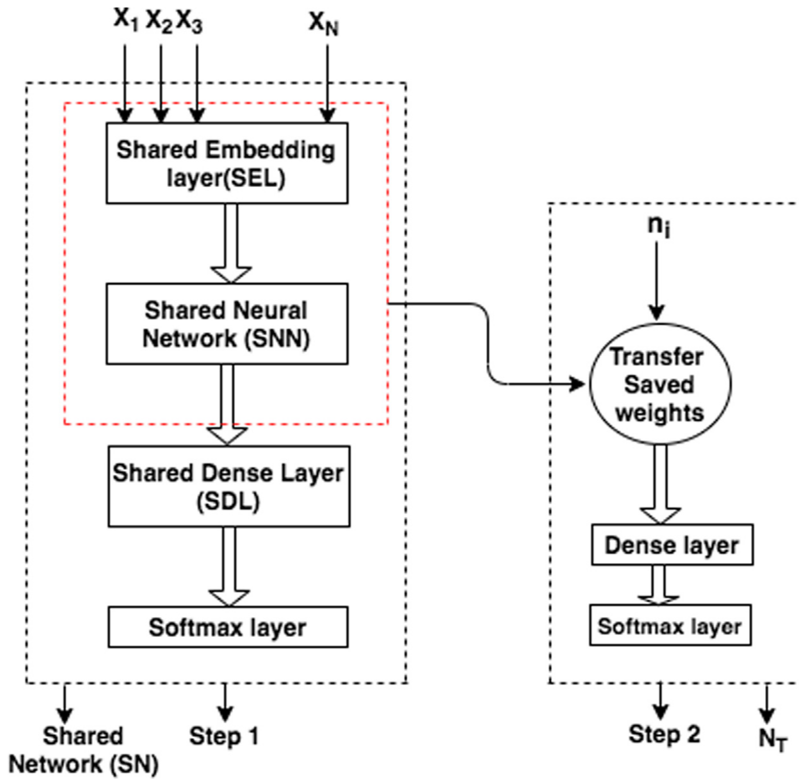

Fig. 6. Architecture of Shared Knowledge Transfer.

Table 31
Table showing misclassifications percentage of one class into other for STL and SP-MTL.

Mode	Class	Misclassification
STL	Hate(D1)	Offensive(60.76%) Neutral(9.2%)
SP-MTL	Hate(D1)	Offensive(23.42%) neutral(4.6%)
STL	Offensive(D1)	Hate(1.9%)Neutral(2.5%)
SP-MTL	Offensive(D1)	Hate(1.1%) Neutral(1.2%)
STL	Racism(D2)	Sexism(0.4%) Neutral(19%)
SP-MTL	Racism(D2)	Sexism(0.4%)Neutral(8.78%)
STL	Sexism(D2)	Racism(0.34%)Neutral(35.66%)
SP-MTL	Sexism(D2)	Racism(0.76%) Neutral(14.03%)
STL	CAG(D3)	OAG(12.98%)NAG(25.44%)
SP-MTL	CAG(D3)	OAG(5.96%)NAG(9.81%)
STL	OAG(D3)	CAG(50.07%)NAG(11.61%)
SP-MTL	OAG(D3)	CAG(10.73%)NAG(6%)
STL	Offensive(D4)	NOT(38.56%)
SP-MTL	Offensive(D4)	NOT(12.79%)
STL	Harassment(D5)	NON(77.18%)
SP-MTL	Harassment(D5)	NON(10.65%)

misclassification percentage for all the class to another class is given for both Single task learning (STL) and Multi-task learning (MTL) model.

Qualitative Analysis: Tables 32 and 33 enlist some of the posts originally tagged as class *a* mis-classified into class *b* using STL. However SP-MTL system correctly classified these posts into class *a*. Words like fag, faggot that are used to marginalize any group is captured in MTL to be hate in S1 and S2.[13] Indirect reference to an animal in a derogatory manner is correctly classified in S8 and S15. S11 correctly identifies the sexist tone. The mention of racist terms in S25 and S26 are correctly classified to harassment class.

8. Conclusion and future work

The dissemination of hateful speech has increased in recent years and has posed a lot of challenges in discriminating hate speech against the freedom of speech. There is no universally accepted definition of hate speech, and this has led to various benchmarking, depending upon the demography, social influences, and cultural factors. Although there exist a few datasets related to either hate or its aligned tasks, such as racism, offensive language identification, etc, these are often limited by the size and do not follow any uniform annotation schema. In this paper, we have proposed a deep multi-task learning framework to leverage useful information from these multiple related tasks. Experiments performed on five benchmark datasets show the efficacy of the proposed approach that attains convincing performance over the state-of-the-art models. The Evaluation also demonstrates that the proposed multi-task model achieves better performance compared to the single-tasking framework.

From all the results it can be inferred that different data set classified in various subclasses do indeed help each other in classification. So instead of creating new data set and annotating it with another set of classes (which could overlap to the other existing ones), we emphasize on creating data classified into *Offensive* and *Non-Offensive*. These two classes can be further classified into subclasses according to the need as in [21,83,89]. All the non-offensive posts will be allowed as the right to freedom of speech, whereas the offensive post can be further studied and annotated to understand its sentiment.

The future work will attempt towards creating a more fine-grained multi-layer annotation schema and explore domain-specific embedding. The addition of user meta information, user behavior in posting harmful speech, probability of diffusion of any harmful text as a feature can be incorporated into the model.

So far this problem is mostly been modeled as a supervised task, but unsupervised learning can effectively be used to leverage

[13] The example number in Tables 32 and 33 are prefixed with *S*

Table 32
Examples of original post misclassified in STL and correctly classified in SP-MTL.

Original	STL	SP-MTL	Tweet
1.Hate	Offensive	Hate	@AGreasyMonkey @Huntermoore fa**ot
2.Hate	Offensive	Hate	@DevilFzyq @DevilElyie @TehDevilClan shut up f*g
3.Hate	Neutral	Hate	,@KiroGx me gusta "typing like a retard to make the other person look dumber by pretending to agree with them" bazinga
4.Hate	Neutral	Hate	@ReiddH i agree my fellow cracker.
5.Offensive	Hate	Offensive	@Juice_2 nah you're a ni**er
6.Offensive	Hate	Offensive	@Kvrmaa_Trill @StudyFN @VLzed says the scrawny ni**er.
7.Offensive	Neutral	Offensive	@4REALROY: @Vomitdatruth no more chunky nugs for you; he doesn't even hook it up like that.
8.Offensive	Neutral	Offensive	@DecodnLyfe @LupeFiasco @Larellj another black man? What does that have to do with anything? Once a monkey, always a monkey, Chicago idiot.
9.Racism	Neutral	Racism	@ardiem1m @Alfonso_AraujoG @MaxBlumenthal @oldkhayyam Yeah, being anti Zionist is pretending to have an excuse for being anti Semitic.
10.Racism	Neutral	Racism	@OneLegSandpiper @DblBlackDs Looks like you are the ignorant a*****e, in that there is no other religion that has close to that number.
11.Sexism	Neutral	Sexism	SO HILARIOUS U WRITE UR OWN MATERIAL? @JesseElJefe A lot of ppl call me sexist. But those ppl are women, and their opinions don't matter.
12. Sexism	Neutral	Sexism	RT @EBeisner @ahall012 I agree with you!! I would rather brush my teeth with sandpaper then watch football with a girl!!.
13.OAG	CAG	OAG	Well you pour money for medals but no money for education of poors shame on govt
14.OAG	CAG	OAG	Poor r starving n people r wasting thousand litres of milk on d posters of their favourite actors coz they treat them like God.... It's nt foolishness or stupidity. It's disgusting
15.OAG	NAG	OAG	she is screaming like a pig went under the tyre.
16.OAG	NAG	OAG	Most irresponsible ex cricketer of world.

Table 33
Examples of original post misclassified in STL and correctly classified in SP-MTL.

Original	STL	SP-MTL	Tweet
17.CAG	NAG	OAG	absolutely! the deeper you dive the shallower cushion you have.
18.CAG	NAG	OAG	Now he is going to be a headache for his co-passengers during the train journey.
19.CAG	OAG	CAG	There is no need to give a statement after doing such a bit of work. You know what kind of person you are. You can not even find a place in hell.
20.CAG	OAG	CAG	not only makes a mockery of all Delhites who voted him but also showed how to misuse his power ...all the best coz ur b lame game is ur only initiative u hv taken ever since u became cm ...crap !!!
21.OFF	NON	OFF	@USER Ohhhhhh I cried BIG crocodile tears the first time my daddy called me a b**ch took it on the chin like a champ but ran to my room to cry just like a b**ch
22.OFF	NON	OFF	@USER Fuckin poisonous group of people. As soon as someone hands me a hat with a propeller on top... I'm out. Find another company to work for lemmings.
23. NON	OFF	NON	@USER @USER He is delusional.
24. NON	OFF	NON	@USER @USER She is flat out lying!.
25.HAR	NOT	HAR	@BunkerHillD antiwhites all have cancer of the soul WhiteGenocide
26.HAR	NOT	HAR	@Curie_The_Ai gas the Jews
27.NON	HAR	NON	I did not say gas the jews, I said glass of juice!
28.NON	HAR	NON	@Maarvn cause you hate the Jews

the large amount of social media data available. Meta-heuristics algorithms for selecting the best fitting feature combinations under the supervised setting may be explored. For this purpose, we plan to explore the following techniques: Multi-verse optimizer [90], Group search optimizer [91], Harmony search optimizer [92], Krill herd algorithm [93–96], Hybrid antlion optimization [97], Particle swarm optimization [98,99], and Genetic algorithm [100].

CRediT authorship contribution statement

Prashant Kapil: Methodology design and Implementation, Writing - original draft, Data curation, Experiments, Analysis. **Asif Ekbal:** Conceptualization, Methodology, Writing - review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first author would like to acknowledge the funding agency, the University Grant Commission (UGC) of the Government of India, for providing financial support in the form of UGC NET-JRF/SRF. Asif Ekbal acknowledges the Young Faculty Research Fellowship (YFRF), supported by Visvesvaraya Ph.D. scheme for Electronics and IT, Ministry of Electronics and Information Technology (MeitY), Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

References

[1] Benesch Susan, Countering dangerous speech to prevent mass violence during Kenya's 2013 elections, Final Report, 2014, pp. 1–26.
[2] Davidson Thomas, Dana Warmsley, Michael Macy, Ingmar Weber, Weber automated hate speech detection and the problem of offensive language, 2017, arXiv preprint arXiv:1703.04009.
[3] J.T. Nockleby, 'Hate speech in encyclopedia of the American constitution, 2000.
[4] Waseem Zeerak, Dirk Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in: Proceedings of the NAACL student research workshop, 2016, pp. 88-93.
[5] Cohen-Almagor Raphael, Fighting hate and bigotry on the internet, Policy Internet 3 (3) (2011) 1–26.
[6] Warner William, Julia Hirschberg, Detecting hate speech on the world wide web, in: Proceedings of the second workshop on language in social media, 2012, pp. 19-26.
[7] Nobata Chikashi, Joel Tetreault, Achint Thomas, Yashar Mehdad, Yi Chang, Abusive language detection in online user content, in: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 145-153.
[8] Badjatiya Pinkesh, Shashank Gupta, Manish Gupta, Vasudeva Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017.
[9] Bojanowski Piotr, Edouard Grave, Joulin Armand, Tomas Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146.
[10] Pennington Jeffrey, Richard Socher, Christopher D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543.
[11] Mehdad Yashar, Joel Tetreault, Do characters abuse more than words? in: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2016, pp. 299-303.
[12] Zhang Ziqi, David Robinson, Jonathan Tepper, Detecting hate speech on twitter using a convolution-gru based deep neural network, in: European Semantic Web Conference, Springer, Cham, 2018, pp. 745–760.
[13] Mikolov Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, Jeff Dean, Distributed representations of words and phrases and their compositionality, Adv. Neural Inform. Process. Syst. (2013) 3111–3119.
[14] Gambäck Björn, Utpal.Kumar. Sikdar, Using convolutional neural networks to classify hate-speech, in: Proceedings of the First Workshop on Abusive Language Online, 2017, pp. 85-90.
[15] Waseem Zeerak, Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter, in: Proceedings Of the First Workshop on NLP and Computational Social Science, 2016, pp. 138-142.
[16] MacAvaney Sean, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, Ophir Frieder, Hate speech detection: Challenges and solutions, PLoS One 14 (8) (2019) e0221152.
[17] Devlin Jacob, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[18] Ona de Gibert, Naiara Perez, Aitor García-Pablos, Montse Cuadros, Hate speech dataset from a white supremacy forum, 2018, arXiv preprint arXiv:1809.04444.
[19] Pérez Juan Manuel, Franco M. Luque, Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 64-69.
[20] Matthew E Peters, Neumann Mark, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, Deep contextualized word representations, 2018, arXiv preprint arXiv:1802.05365.
[21] Basile Valerio, Cristina Bosco, Elisabetta Fersini, Nozza Debora, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, Manuela Sanguinetti, Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter, in: 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2019, pp. 54–63.
[22] Gitari Njagi Dennis, Zhang Zuping, Hanyurwimfura Damien, Jun Long, A lexicon-based approach for hate speech detection, Int. J. Multimedia Ubiq. Eng. 10 (4) (2015) 215–230.
[23] Amir H. Razavi, Diana Inkpen, Sasha Uritsky, Stan Matwin, Offensive language detection using multi-level classification, in: Canadian Conference on Artificial Intelligence, Springer, Berlin, Heidelberg, 2010, pp. 16–27.
[24] Ian H Witten, Eibe Frank, Data mining: practical machine learning tools and techniques with java implementations, Acm Sigmod Rec. 31 (1) (2002) 76–77.
[25] Wiegand Michael, Josef Ruppenhofer, Anna Schmidt, Clayton Greenberg, Inducing a lexicon of abusive words–a feature-based approach, 2018.
[26] Wilson Theresa, Janyce Wiebe, Paul Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 2005, pp. 347-354.
[27] Brassard-Gourdeau Eloi, Richard Khoury, Subversive toxicity detection using sentiment information, in: Proceedings of the Third Workshop on Abusive Language Online, 2019, pp. 1-10.
[28] Joksimovic Srecko, Ryan S Baker, Jaclyn Ocumpaugh, L. Andres Juan Miguel, Ivan Tot, Elle Yuan Wang, Shane Dawson, Automated identification of verbally abusive behaviors in online discussions, in: Proceedings of the Third Workshop on Abusive Language Online, 2019, pp. 36-45.
[29] Scott A Crossley, Kristopher Kyle, Danielle S. McNamara, The tool for the automatic analysis of text cohesion (TAACO) automatic assessment of local, global, and text cohesion, Behav. Res. Methods 48 (4) (2016) 1227–1237.
[30] C.H.E Gilbert, Erric Hutto, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: Eighth International Conference on Weblogs and Social Media (ICWSM-14), 2014, pp. 81–82, Available at (20/04/16) http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf.
[31] Chatzakou Despoina, Nicolas Kourtellis, Jeremy Blackburn, Emiliano.De Cristofaro, Gianluca Stringhini, Athena Vakali, Mean birds: Detecting aggression and bullying on twitter, in Proceedings of the 2017 ACM oN Web Science Conference, 2017, pp. 13-22.
[32] Qian Jing, Mai ElSherief, Elizabeth M. Belding, Yang Wang William, Leveraging intra-user and inter-user representation learning for automated hate speech detection, 2018), arXiv preprint arXiv:1804.03124.
[33] Indyk Piotr, Rajeev Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: Proceedings Of the Thirtieth Annual ACM Symposium on Theory of Computing, 1998, pp. 604-613.
[34] Papegnies Etienne, Vincent Labatut, Richard Dufour, Georges Linares, Graph-based features for automatic online abuse detection, in: International Conference on Statistical Language and Speech Processing, Springer, Cham, 2017, pp. 70–81.
[35] Gröndahl Tommi, Luca Pajola, Mika Juuti, Mauro Conti, N. Asokan, All you need is love evading hate speech detection, in: Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, 2018, pp. 2-12.
[36] Vidgen Bertie, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, Helen Margetts, Challenges and Frontiers in Abusive Content Detection, Association for Computational Linguistics, 2019.
[37] Founta Antigoni-Maria, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, Nicolas Kourtellis, Large scale crowdsourcing and characterization of twitter abusive behavior, 2018, arXiv preprint arXiv:1802.00393.
[38] Fortuna Paula, Sérgio Nunes, A survey on automatic detection of hate speech in text, ACM Comput. Surv. 51 (4) (2018) 1–30.
[39] Founta Antigoni Maria, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, Ilias Leontiadis, A unified deep learning architecture for abuse detection, in: Proceedings of the 10th ACM Conference on Web Science, 2019, pp. 105-114.
[40] Karan Mladen, Jan Šnajder, Cross-domain detection of abusive language online, in: Proceedings of the 2nd workshop on abusive language online (ALW2), 2018, pp. 132-137.

[41] Hal Daumé III, Frustratingly easy domain adaptation, 2009, arXiv preprint arXiv:0907.1815.
[42] Wiegand Michael, Josef Ruppenhofer, Thomas. Kleinbauer, Detection of abusive language: the problem of biased datasets, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), 2019, pp. 602-608.
[43] Bolukbasi Tolga, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, Adam T. Kalai, Man is to computer programmer as woman is to homemaker? debiasing word embeddings, in Advances in neural information processing systems, 2016, pp. 4349-4357.
[44] Davidson Thomas, Debasmita Bhattacharya, Ingmar. Weber, Racial bias in hate speech and abusive language detection datasets, 2019, arXiv preprint arXiv:1905.12516.
[45] Ji Ho Park, Jamin Shin, Pascale. Fung, Reducing gender bias in abusive language detection, 2018, arXiv preprint arXiv:1808.07231.
[46] Chauhan Dushyant Singh, Rohan Kumar, Asif Ekbal, Attention based shared representation for multi-task stance detection and sentiment analysis, in: International Conference on Neural Information Processing, Springer, Cham, 2019, pp. 661–669.
[47] Deep Kumar Shikhar, Md Shad Akhtar, Asif Ekbal, Pushpak Bhattacharyya, Related tasks can share! a multi-task framework for affective language, 2020, arXiv preprint arXiv:2002.02154.
[48] Sangwan Suyash, Dushyant Singh Chauhan, Md Shad Akhtar, Asif Ekbal, Pushpak Bhattacharyya, Multi-task gated contextual cross-modal attention framework for sentiment and emotion analysis, in: International Conference on Neural Information Processing, Springer, Cham, 2019, pp. 662–669.
[49] Yadav Shweta, Asif Ekbal, Sriparna Saha, Pushpak. Bhattacharyya, A unified multi-task adversarial learning framework for pharmacovigilance mining, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5234-5245.
[50] Yu Zhang, Qiang Yang, A survey on multi-task learning, 2017, arXiv preprint arXiv:1707.08114.
[51] Xue Ya, Xuejun Liao, Lawrence Carin, Balaji Krishnapuram, Multi-task learning for classification with dirichlet process priors, J. Mach. Learn. Res. 8 (Jan.) (2007) 35–63.
[52] Ruder Sebastian, An overview of multi-task learning in deep neural networks, 2017, arXiv preprint arXiv:1706.05098.
[53] Kim Yoon, Convolutional neural networks for sentence classification, 2014, arXiv preprint arXiv:1408.5882.
[54] Collobert Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (Aug) (2011) 2493–2537.
[55] Hochreiter Sepp, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[56] Liu Pengfei, Xipeng Qiu, Xuanjing. Huang, Adversarial multi-task learning for text classification, 2017, arXiv preprint arXiv:1704.05742.
[57] Kumar Ritesh, Aishwarya N. Reganti, Akshi Bhatia, Tushar Maheshwari, Aggression-annotated corpus of hindi-english code-mixed data, 2018, arXiv preprint arXiv:1803.09402.
[58] Zampieri Marcos, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, Ritesh Kumar, Predicting the type and target of offensive posts in social media, 2019, arXiv preprint arXiv:1902.09666.
[59] Golbeck Jennifer, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, et al., A large labeled corpus for online harassment research, in: Proceedings of the 2017 ACM On Web Science Conference, 2017, pp. 229-233.
[60] Djuric Nemanja, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, Narayan Bhamidipati, Hate speech detection with comment embeddings, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 29-30.
[61] William J. Wilson, The Bridge over the Racial Divide: Rising Inequality and Coalition Politics. Vol. 2, Univ of California Press, 1999.
[62] Glick Peter, Susan T. Fiske, Ambivalent sexism revisited, Psychol. Women Quart. 35 (3) (2011) 530–535.
[63] Benatar David, The Second Sexism: Discrimination Against Men and Boys, John Wiley & Sons, 2012.
[64] Robert A. Baron, Deborah R. Richardson, Human Aggression, Springer Science & Business Media, 2004.
[65] Hardaker Claire, Trolling in asynchronous computer-mediated communication: From user discussions to academic definitions, J. Polit. Res. 6 (2) (2010) 215–242.
[66] Waseem Zeerak, Thomas Davidson, Dana Warmsley, Ingmar Weber, Understanding abuse: A typology of abusive language detection subtasks, 2017, arXiv preprint arXiv:1705.09899.
[67] Chollet François, Keras, 2015, https://keras.io.
[68] Abadi Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al., Tensorflow: Large-scale machine learning on heterogeneous systems, 2015.
[69] Chakrabarty Tuhin, Kilol Gupta, Smaranda Muresan, Pay attention to your context when classifying abusive language, in: Proceedings of the Third Workshop on Abusive Language Online, 2019, pp. 70-79.
[70] Betty van Aken, Julian Risch Ralf Krestel, Alexander Löser, Challenges for toxic comment classification: An in-depth error analysis, 2018, arXiv preprint arXiv:1809.07572.
[71] Naseem Usman, Imran Razzak, Ibrahim A. Hameed, Deep context-aware embedding for abusive and hate speech detection on twitter, Aust. J. Intell. Inf. Process. Syst. 15 (3) (2019) 69–76.
[72] Mishra Pushkar, Helen Yannakoudakis, Ekaterina Shutova, Neural character-based composition models for abuse detection, 2018, arXiv preprint arXiv:1809.00378.
[73] Kshirsagar Rohan, Tyus Cukuvac, Kathleen McKeown, Susan McGregor, Predictive embeddings for hate speech detection on twitter, 2018, arXiv preprint arXiv:1809.10644.
[74] Ji Ho Park, Pascale Fung, Fung one-step and two-step classification for abusive language detection on twitter, 2017, arXiv preprint arXiv:1706.01206.
[75] Meyer Johannes Skjeggestad, Björn Gambäck, A platform agnostic dual-strand hate speech detector, in: ACL 2019 the Third Workshop on Abusive Language Online Proceedings of the Workshop, Association for Computational Linguistics, 2019.
[76] Steimel Kenneth, Daniel Dakota, Yue Chen, Sandra Kübler, Investigating multilingual abusive language detection: a cautionary tale, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), 2019, pp. 1151-1160.
[77] Georgios K Pitsilis, Heri Ramampiaro, Helge Langseth, Effective hate-speech detection in twitter data using recurrent neural networks, Appl. Intell. 48 (12) (2018) 4730–4742.
[78] Prashant Kapil, Asif Ekbal, Dipankar Das, Investigating deep learning approaches for hate speech detection in social media, 2020, arXiv preprint arXiv:2005.14690.
[79] Zhang Xiang, Junbo Zhao, Yann LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.
[80] Liu Ping, Wen Li, Liang. Zou, 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, 2019, pp. 87-91.
[81] Cambray Aleix, Norbert Podsadowski, Bidirectional recurrent models for offensive tweet classification, 2019, arXiv preprint arXiv:1903.08808.
[82] Marwa Tolba, Ouadfel Salima, Meshoul Souham, Deep learning for online harassment detection in tweets, in: 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), IEEE, 2018, pp. 1–5.
[83] Zampieri Marcos, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, Ritesh Kumar, Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval), 2019, arXiv preprint arXiv:1903.08983.
[84] Srivastava Saurabh, Prerna. Khurana, Detecting aggression and toxicity using a multi dimension capsule network, in: Proceedings of the Third Workshop on Abusive Language Online, 2019, pp. 157-162.
[85] Segun Taofeek Aroyehun, Gelbukh Alexander, Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 90-97.
[86] Arroyo-Fernández Ignacio, Dominic Forest, Juan-Manuel Torres-Moreno, Mauricio Carrasco-Ruiz, Thomas Legeleux, Karen Joannette, Cyberbullying detection task: the ebsi-lia-unam system (elu) at coling'18 trac-1, in: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), 2018, pp. 140-149.
[87] Zhu Jian, Zuoyu Tian, Sandra Kübler, Um-iu@ ling at semeval-2019 task 6: Identifying offensive tweets using bert and svms, 2019, arXiv preprint arXiv:1904.03450.
[88] Pelicon Andraž, Matej Martinc, Petra.Kralj. Novak, Embeddia at SemEval-2019 Task 6: Detecting hate with neural network and transfer learning approaches, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 604-610.
[89] Mandl Thomas, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, Aditya. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019, pp. 14-17.
[90] Abualigah Laith, Multi-verse optimizer algorithm: a comprehensive survey of its results variants and applications, Neural Comput. Appl. (2020) 1–21.
[91] Abualigah Laith, Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications, Neural Comput. Appl. (2020) 1–24.
[92] Abualigah Laith, Ali Diabat, Zong Woo Geem, A comprehensive survey of the harmony search algorithm in clustering applications, Appl. Sci. 10 (11) (2020) 3827.

[93] Laith Mohammad Abualigah, Ahamad Tajudin Khader, Essam Said Hanandeh, Hybrid clustering analysis using improved krill herd algorithm, Appl. Intell. 48 (11) (2018) 4047–4071.
[94] Laith Mohammad Qasim Abualigah, Feature Selection and Enhanced Krill Herd Algorithm for Text Document Clustering, Springer, Berlin, 2019.
[95] Abualigah Laith Mohammad, Ahamad Tajudin Khader, Essam Said Hanandeh, A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis, Eng. Appl. Artif. Intell. 73 (2018) 111–125.
[96] Laith Mohammad Abualigah, Ahamad Tajudin Khader, Essam Said Hanandeh, Amir H. Gandomi, A novel hybridization strategy for krill herd algorithm applied to clustering techniques, Appl. Soft Comput. 60 (2017) 423–435.
[97] Abualigah Laith, Ali Diabat, A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments, Cluster Comput. (2020) 1–19.
[98] Abualigah Laith Mohammad, Ahamad Tajudin Khader, Essam Said Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466.
[99] Laith Mohammad Abualigah, Ahamad Tajudin Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, J. Supercomput. 73 (11) (2017) 4773–4795.
[100] Laith Mohammad Qasim Abualigah, Essam S. Hanandeh, Applying genetic algorithms to information retrieval using vector space model, Int. J. Comput. Sci. Eng. Appl. 5 (1) (2015) 19.