Roman Urdu toxic comment classification

Language Resources and Evaluation manuscript No.
(will be inserted by the editor)

Roman Urdu Toxic Comment Classification

Hafiz Hassaan Saeed · Muhammad Haseeb Ashraf · Faisal Kamiran · Asim Karim · Toon Calders

Received: date / Accepted: date

Abstract With the increasing popularity of user-generated content on social media, the number of toxic texts is also on the rise. Such texts cause adverse effects on users and society at large, therefore, the identification of toxic comments is a growing need of the day. While toxic comment classification has been studied for resource-rich languages like English, no work has been done for Roman Urdu despite being a widely used language on social media in South Asia. This paper addresses the challenge of Roman Urdu toxic comment detection by developing a first-ever large labeled corpus of toxic and non-toxic comments. The developed corpus, called RUT (Roman Urdu Toxic), contains over 72 thousand comments collected from popular social media platforms and

Hafiz Hassaan Saeed
Information Technology University,
Lahore, Pakistan
E-mail: hassaan.saeed@itu.edu.pk
orcid=0000-0001-5026-0765

Muhammad Haseeb Ashraf
Information Technology University,
Lahore, Pakistan
E-mail: msds18006@itu.edu.pk
orcid=0000-0002-6345-454X

Faisal Kamiran
Information Technology University,
Lahore, Pakistan
E-mail: faisal.kamiran@itu.edu.pk

Asim Karim
Lahore University of Management Sciences,
Lahore, Pakistan
E-mail: akarim@lums.edu.pk

Toon Calders
University of Antwerp,
Antwerp, Belgium
E-mail: toon.calders@uantwerpen.be

has been labeled manually with a strong inter-annotator agreement. With this dataset, we train several classification models to detect Roman Urdu toxic comments, including classical machine learning models with the bag-of-words representation and some recent deep models based on word embeddings. Despite the success of the latter in classifying toxic comments in English, the absence of pre-trained word embeddings for Roman Urdu prompted to generate different word embeddings using Glove, Word2Vec and FastText techniques, and compare them with task-specific word embeddings learned inside the classification task. Finally, we propose an ensemble approach, reaching our best F1-score of 86.35%, setting the first-ever benchmark for toxic comment classification in Roman Urdu.

Keywords Roman Urdu · Toxic Comment Classification · Deep Learning · Roman Urdu Toxic Comments · Deep Ensemble

1 Introduction

The word "Toxic" means "Poisonous" in its literal meaning. Recently, it gained prominence in the context of text data during a famous competition hosted at Kaggle on classifying comments that are rude, disrespectful, or intended to make someone leave a conversation[1]. In this paper, we generalize their notion to the use of hateful or abusive words, obscenity, threat, insult, or execration towards gender, color, race, religion, ethnicity, ideology, culture, etc in a text. Some synonymous terms used interchangeably in the literature for the detection of toxic comments are hate speech [53], cyberbullying [38], abusive language [28], profanity [30], and malicious comments [29]. The uncontrolled spread of online toxic comments or hate speech has emerged as an undesirable global social issue [29], which can even cause its victims to commit suicide. For example, a 12-year-old girl[2] and a famous television host named Charlotte Dawson[3] committed suicide after being victims of cyberbullying and online misogyny. As a result, many social media platforms try to address the growing issue of toxic comments by investing hundreds of millions of euros every year in toxic comments detection and mitigation [17].

While toxic text detection has been studied extensively in English and various other resource-rich languages, Roman Urdu still remains neglected and lacks the attention of the research community despite having a huge user-base. Roman Urdu is a popular romanized style of writing of the Urdu language that uses the characters of the English alphabet. Originally, Urdu is written in Perso-Arabic script and is the national language of Pakistan having more than 170 million speakers worldwide [13]. It is also the official language of many Indian states and is among the widely spoken and understood languages of

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[2] https://www.theguardian.com/world/2013/oct/15/florida-cyberbullying-rebecca-sedwick-two-girls-arrested

[3] https://www.cosmopolitan.com/uk/reports/a25443/charlotte-dawson-dead-suicide-trolling/

South Asia [39]. According to Bilal et al. [9], the reasons for the popularity of Roman Urdu are lack of comfort in English and unavailability of easy-to-use Urdu keyboards for Arabic script. Roman Urdu is an informal and colloquial language that lacks standard word forms or any defined spelling criteria. For example, the word "khubsurat" translated in English as [Beautiful] can also be written as "khoobsoorat", "khobsorat", "khoobsurat", "khubsoorat", "kubsorat", "kubsoret" and "khubsoret". Osama et al. reported that a single word in Roman Urdu can be written in 4 different ways on average and reported to have found the word "mohabbat" translated in English as [Love] in 79 different spelling variations [25]. Language processing in Roman Urdu becomes more complicated because of several words that are lexically different but can be spelled in the same way, for example, the word "chor" can be used for both [thief] and [to leave].

Learning high-performance language models relies primarily on the availability of resources (datasets). Roman Urdu lacks resources despite the few available datasets that are not large in size in general, e.g., Mehmood et al. developed a dataset of 11 thousand reviews for sentiment analysis in Roman Urdu [32]. In addition, neither of the existing datasets is adequate enough for toxic comment detection in Roman Urdu. Therefore, in this paper, we prepare a large and a first-ever corpus for the detection of toxic comments in Roman Urdu. The developed corpus contains 72,771 diverse comments scraped from Facebook, Twitter, YouTube, and a few other websites. We name this collected corpus the *RUT Corpus* where RUT refers to "Roman Urdu Toxic" comments. All comments in the RUT Corpus have been labeled manually into toxic and non-toxic classes by two independent annotators with 0.8191 inter-annotator agreement in terms of the kappa statistic. The disagreement was resolved by adding a third annotator.

In recent years, deep learning based approaches have shown remarkably improved scores for complex tasks of computer vision, speech and language processing as compared to classical machine learning approaches. This paper evaluates existing state-of-the-art techniques developed for text classification over the prepared corpus. More specifically, we compare the performance of four classical machine learning classifiers (Naïve Bayes, Random Forests, Logistic Regression, Support Vector Machines), three state-of-the-art deep learning models proposed for toxic comment classification in English, two variants of Recurrent Neural Networks (Bidirectional LSTM and Bidirectional GRU) and a tweaked Convolutional Neural Network. Most deep learning techniques for text classification problems use word embeddings to represent text. Unlike English where high-quality pre-trained word embeddings are abundantly available, Roman Urdu does not have such pre-trained word-level or character-level embeddings. We, therefore, train several general-purpose word embeddings including standard Glove, Word2Vec and FastText techniques using the prepared corpus, and empirically compare them with task-specific word embeddings. We try both skip-gram and continuous bag-of-words versions of Word2Vec and FastText, whereas by task-specific word embeddings we mean learning the word embedding as part of training a toxic comment classifier.

The experiments reported in this paper are performed using stratified 5-fold cross-validation. We report the averages of accuracy, precision, recall and F1 over all 5 folds. All hyper-parameters are tuned as part of the model learning process and included inside the cross-validation loop. With the trained classification models aforementioned, we use an ensemble approach by testing different combinations of the trained classifiers inside the ensemble classifier.

In summary, the contributions of this paper are:

- The first contribution of this paper is the manual preparation and development of the first-ever corpus for the detection of Roman Urdu toxic comments with a strong inter-annotator agreement. The developed corpus is publicly available to excite and trigger further research.
- The second and foremost contribution of this paper is the detection of toxic text in Roman Urdu, which to the best of our knowledge has never been studied before.
  - We classify given Roman Urdu text into the toxic class or non-toxic class.
  - We compare several pre-trained general-purpose word embedding techniques with task-specific word embeddings integrated in the classification models to circumvent the absence of pre-trained word embeddings for Roman Urdu.
- The third contribution of this paper is an extensive empirical evaluation of a number of learning models over the prepared corpus. The learning models include simple text classification methods and a few recent classification models that are best suited to the classification of toxic comments in English.
- The fourth contribution of this paper is the analysis of classifiers' predictions using Cohen's kappa statistic. Based on the variations in the predictions of the models, we propose an ensemble that improves the scores as compared to the individual models on our prepared corpus.

The rest of the paper is organized as follows: Section 2 discusses the existing work related to the scope of this research study, Section 3 describes the collection and annotation process of the corpus along with the inter-annotator agreement and some corpus statistics, Section 4 describes the overall methodology adapted in this paper and the classification models used, Section 5 discusses the experimental details and the results of the experiments, Section 6 discusses the results and behavior of the proposed ensemble, and, finally Section 7 concludes this study.

2 Related Work

Toxic Comment Classification has been studied for many years in the literature with various other terms like hate speech, cyberbullying, abuse detection, profanity detection, or malicious comments. For the task of toxic text detection, multiple resources (datasets) exist for various languages like English

[8, 12, 14, 18, 43, 35, 51], Italian [12, 16, 37, 45, 50], Spanish [5, 8, 15], Arabic [4, 21], Indonesian [6, 49], French [12], German [42], Dutch [22]. A few of these datasets are very limited in size, for example, [42] contains 541 tweets, [49] contains 2273 comments, [45] contains 6000 comments. To the best of our knowledge, no dataset is available for the detection of toxic text in Roman Urdu. We would like to mention that there are some datasets available for sentiment analysis in Roman Urdu [32, 19] but sentiment analysis is different from toxic text detection. A mere negative sentiment might not be a toxic comment.

The existing works used for toxic comment classification include [2, 18, 24, 35, 43, 44, 47], for hate speech detection include [7, 17, 30, 46, 52, 53], cyberbullying [1, 3, 23, 38, 40], abusive language [28, 34] and profanity [30, 48]. Toxic comment classification is a text classification task where rule-based, machine learning and deep learning approaches have been used. Lee et al. [28] propose a decision system to detect abusive text using unsupervised learning by preparing a list of abusive words based on Word2Vec's skip-gram and cosine similarity.

Reynolds et al. [40] apply several machine learning techniques including decision trees, support vector machines, rule-based and instance-based algorithms available in the WEKA toolkit, to detect language patterns used by bullies and develop rules to detect cyberbullying content automatically. Less than 10% of their dataset is positive (bullying content). To overcome the class imbalance problem, they increase the weights of the positive instances. Hosseinmardi et al. [23] investigate a media-based social network Instagram for the prediction of cyberbullying incidents. Their dataset comprises of media sessions along with user comments. This work proposes building a predictor to anticipate the occurrence of cyberbullying before it actually occurs. They use a logistic regression classifier to train a predictor with the forward feature selection approach. Garadi et al. [3] propose supervised machine learning solutions including naïve bayes, support vector machine, random forest, and k-nearest neighbor based on multiple unique features derived from the data. They use data from Twitter and use synthetic minority oversampling technique (SMOTE) and the weight adjusting approach to balance the dataset because of the extremely low number of cyberbullying data. Shtovba et al. [47] show that syntactic dependencies in sentences affect the quality of detecting toxic comments on social networks. They propose three additional features and use decision trees as classifier. Watanabe et al. [52] propose an approach that is based on unigrams and patterns which are used as features to train machine learning algorithms that include support vector machine, random forest and J48graft. They classify a tweet into three different classes as hateful, offensive or clean.

Ptaszynski et al. [38] use multiple classifiers such as naïve bayes, support vector machines, k-nearest neighbours, random forest, J48, JRip, Sentence Pattern Extraction arChitecture (SPEC) and convolution neural network (CNN) for cyberbullying detection and the results show that CNN beats different classifiers by over 11% in F-score. Badjatiya et al. [7] compare multiple machine

learning classifiers such as random forests, support vector machines, gradient boosted decision trees, logistic regression, and deep neural networks with Glove word embeddings for hate speech detection in tweets. Glove [36] is a log-bilinear regression model for unsupervised learning of vector representations for words. Georgakopoulos et al. [18] propose a CNN architecture and compare its performance with multiple machine learning models. Their results show that CNN outperforms well-established machine learning models in toxic comment classification. Zhang et al. [53] use a few pre-processing steps and propose a network architecture combining convolution neural network and gated recurrent unit (CNN+GRU).

Recurrent neural network (RNN) presented in [27] and Bidirectional Long-Short Term Memory (BLSTM) presented in [54] also show an effective performance in text classification tasks. Santosh et al. [46] use sub-word level LSTM and hierarchical LSTM with attention for hate speech detection on social media code-mixed text, based on the phonemic sub-words and show that hierarchical LSTM model with attention, gives good recall and f1-score. Ibrahim et al. [24] propose an ensemble of three learning models that include CNN, BLSTM and BGRU. The proposed technique divides the prediction into two steps and achieves a good f1-score outperforming other methods. The dataset used is highly imbalanced so different data augmentation techniques are used to overcome the class imbalance problem. Risch et al. [41] propose a data augmentation technique that triples the training data. They propose a BGRU and three logistic regression classifiers with hand-picked features, forming an ensemble classifier that gives more robust results. In a few other works, Aken et al. [2] draws a comparison between different deep learning and shallow learning approaches and proposes an ensemble that outperforms each of the individual models. The word embeddings used were based on Glove and FastText, and the individual models include logistic regression, CNN, LSTM, BLSTM, BGRU, and BGRU with attention. Guggilla et al. [20] use two supervised deep learning approaches that include CNN and LSTM for classifying online user comments using Word2Vec and linguistic embeddings. Both CNN and LSTM show significantly improved results over machine learning classifiers like naïve bayes and support vector machines. Agrawal et al. [1] use four deep learning models namely CNN, LSTM, BLSTM and BLSTM with attention to detect cyberbullying across different social media. During their training process, they also apply the concept of transfer learning on different datasets related to different social media platforms.

3 The Dataset - RUT Corpus

Roman Urdu is deficient in language resources [31], especially for toxic text detection. The few corpora available for Roman Urdu in the literature are for Roman Urdu standardization [39], sentiment analysis [32] and transliteration (Roman Urdu to Urdu and vice versa) [10]. None of these corpora are adequate enough for the task of toxic comment classification in Roman Urdu. To the

best of our knowledge, this is the first time that a dataset is prepared and developed for detecting Roman Urdu toxic comments. We now describe the data acquisition and annotation process along with the inter-annotator agreement and corpus characteristics.

Table 1 Inter-annotator ratings matrix

Annotator Ratings		**B**	
		Toxic	Non-Toxic
A	Toxic	12063	1956
	Non-Toxic	2161	56591

3.1 Data Acquisition

The data was collected from various sources like comments under YouTube videos, posts from Facebook, tweets from Twitter, hamariweb.com and a few other Roman Urdu websites. For each of the mentioned sources, respective scrappers were used to acquire the data. We searched several posts, conversations and videos to collect comments on various controversial topics covering politics, celebrities, sportspeople, talk/morning shows, Indo-Pak wars, religions (e.g., Islam, Christianity, Judaism, Atheism, Buddhism, Sikhism, Hinduism, etc.), sectarianism (e.g., ahmadi non-muslims, sunni-shia disputes, etc.), ethnic groups (e.g., balochi, punjabi, pakhtun, sindhi, bengali, etc.) and a few other topics like terrorism, jihad, women/LGBT rights, etc.

Most of this gathered data was found to be written entirely in English, Urdu, or Roman Urdu. We manually discarded the comments written solely in English or solely in Urdu to keep only the Roman Urdu comments in the dataset. However, a few of the remaining Roman Urdu comments in the corpus had some words from English because of the code-switching capability of the original Urdu language. For example, "i hate moslims kyun k ye sab k sab dehshat gard hain" translated in English as [I hate Muslims because all of them are terrorists]. Here "i" and "hate" are English words but the comment (as a whole) is not discarded because it has a substantial proportion of words from Roman Urdu.

3.2 Data Annotation

The entire annotation process carried out in this study is addressed in this section. The comments in RUT Corpus are annotated using binary labels, i.e., a comment is labeled as either toxic or non-toxic. We manually label the gathered RUT Corpus using two annotators where both annotators were independent of each other, i.e., each annotator separately labeled the comments without the influence of the other annotator. For inter-annotator agreement between the two annotators we use Kappa coefficient, k, by using equation 1:

Table 2 Corpus annotation guidelines

Classes	Annotation Guidelines
Toxic	A comment belongs to the "Toxic" class if it holds any one (or more) of the following: – **Abuse**: If a comment contains any abusive word about any individual or a group of people or any other (living or non-living) thing, for example, "vo to aik kutti ka bacha hai jis ki tum baat kar rhe ho" translated in English as [He is one son of a bitch whom you are talking about]. – **Obscene**: If a comment contains nudity or vulgarity in it, for example "teri gaand mai apna lun dalna hai mujhe" translated in English as [I want to put my dick in your ass]. – **Threat**: If a comment contains any threat to any individual or a group of people or any other (living or non-living) thing, for example, "agar tu apne ghar se nikli to main tujhe qatal kar duga" translated in English as [I will kill you if you step out of your house]. – **Insult**: If a comment contains insult to any individual or a group of people or any other (living or non-living) thing, for example, "oye bander ki shakal waley tu yahan kya kar raha hai?" translated in English as [Oh you monkey face, what are you doing here?]. – **Identity Hate**: If a comment contains any type of identity-based hate such as targeting ethnicity, ideology, religion, affiliation, gender, color, race, etc, for example, "ye sab mosalman to dehshatgird hain" translated in English as [All these Muslims are terrorists].
Non-Toxic	A comment belongs to the "Non-Toxic" class if it does not hold any of the characteristics described for the toxic class, for example, "apka comment parh k khushi hui k aaj b ache log hain is mulk mein reh rahe hain" translated in English as [I am happy to read your comment that there are still good people living in this country]. It should be noted that we are labeling comments for toxic text detection instead of simple sentiment analysis. Hence, a simple negative opinion will be labeled as non-toxic unless it holds any toxic characteristics. For example, "nokia 3310 acha fone nahi hai" translated in English as [Nokia 3310 is not a good mobile phone], this comment shall be assigned the non-toxic label.

$$k = \frac{\mathcal{P}_o - \mathcal{P}_e}{1 - \mathcal{P}_e} \tag{1}$$

Here, $\mathcal{P}_o$ is the relative observed agreement between the two independent annotators and $\mathcal{P}_e$ is the hypothetical probability of random agreement between the annotators, i.e., the probability that both annotators agree on either toxic label or non-Toxic label. From the Table 1, $\mathcal{P}_o = (12063+56591)/72771 = 0.9434$ and $\mathcal{P}_e = 0.03765 + 0.64954 = 0.6872$. Putting $\mathcal{P}_o$ and $\mathcal{P}_e$ in equation 1, we get $k = 0.8191$. Hence, the inter-annotator agreement in terms of Kappa coefficient is 0.8191.

The problem of the conflicting labels assigned by the two annotators is addressed by adding a third annotator. The verdict of the third annotator is

considered to be the final decision about the label of the disputed comment. All three annotators were: a) males and aged 20-30; b) having a Masters degree; c) having Urdu as first language (native Urdu speakers) and English as second language; d) well familiar with Roman Urdu writing style. The overall corpus annotation guidelines provided to the data annotators are given in Table 2. Additionally, a clarification and discussion session was organized to establish a clear understanding of the annotation task.

3.3 Corpus Characteristics

The labeled RUT corpus contains a total of 72,771 manually labeled comments for the task of toxic text detection in Roman Urdu. The number of toxic comments is 13,097 whereas the number of non-toxic comments is 59,674 which shows that ∼18% of the corpus contains toxic comments and ∼82% contains non-toxic comments. Thus, there is a class imbalance in the developed corpus. Other corpus-specific statistics are shown in Table 3.

Table 3 Statistics of RUT Corpus

Class	Number of Comments	Maximum Length	Average Length	Vocabulary Size	Total Tokens
Toxic	13097	139	13.82 ± 14.96	25366	181086
Non-Toxic	59674	194	20.73 ± 18.85	78775	1237494
Total	72771	194	19.49 ± 18.40	91244	1418580

3.4 Ethical Aspects

Information privacy is an important global concern to protect the privacy of individuals. In the data acquisition phase, we did not collect any information related to the identity of the individuals who wrote the comments to preserve their individual privacy.

4 Toxic Comment Classification in Roman Urdu

With the labeled RUT corpus, we address the problem of detecting toxic comments in Roman Urdu as a supervised binary text classification problem. Inspired by work on the classification of toxic comments in English, we try classical machine learning methods based on bag-of-words representation and deep learning models based on word embeddings representation, alongside a few comment pre-processing steps.

4.1 Comment Pre-processing

Multiple strategies used to pre-process text data in English include lemmatization, stemming and conversion of all words to lower case. Researchers also remove stop words, punctuation marks, non-printable characters, extra white-spaces, emoticons, URLs, hashtags, mentions and sometimes date-time are also removed. Words having alpha-numeric characters are also cleaned sometimes by removing numeric characters from the words in the text pre-processing phase.

In this paper, we pre-process the comments as: a) all words are converted to lower case; b) non-printable characters, extra white-spaces and punctuation marks are removed. Given the scarcity of resources in Roman Urdu, it is not possible to remove stop words or do lemmatization/stemming.

4.2 Comment Encoding

Learning models whether supervised, semi-supervised or unsupervised cannot take text data directly as input. The text data is transformed into some numerical encoding. Some common and classical encoding schemes include bag-of-words (like TF.IDF), Document Co-occurrence Matrix, etc. The most recent and state-of-the-art text encoding technique is Word Embeddings like Word2Vec trained by Google [33], Glove trained by Stanford [36] and FastText trained by Facebook [11]. The word embeddings technique replaces each word in the given comment by a real-valued word vector, hence, each comment becomes a matrix $E \in \mathbb{R}^{m \times f}$, where m is the length (number of words) of the comment and f is the size of the embedding vector for each word in the corpus.

In this paper, the comments encoded for machine learning models use bag-of-words approach with TF.IDF weights whereas input to deep models is encoded using word embeddings. To the best of our knowledge, no pre-trained word-level or character-level embeddings exist for Roman Urdu, therefore, we propose two alternatives:

- We train general-purpose Glove, Word2Vec and FastText models one-by-one for the generation of word embeddings for Roman Urdu. These models are subsequently used to represent the comments in numerical format allowing to use existing state-of-the-art deep learning architectures for toxic comment detection. For Word2Vec and FastText, we train both continuous bag-of-words (CBOW) and skip-gram (SG) models separately.
- Alternatively, we avoid the additional step of learning a general-purpose word embedding for Roman Urdu by slightly altering the models. We extend the models by learning the word embeddings as part of the learning process of the models, this way task-specific embedding is learned for Roman Urdu toxic comment classification.

4.3 State-of-the-art Methods in English

We employ several existing state-of-the-art deep models that are developed for classification of toxic comments in English, over the labeled RUT corpus. The deep models are either based on convolutional networks, recurrent networks, or both. Besides the deep models, we also experiment classical machine learning models including Naïve Bayes, Random Forests, Logistic Regression and Support Vector Machines with TF.IDF weights. We now briefly describe the architectures of the deep models used.

4.3.1 CNN-George

CNN-George is the best model reported in [18] used for binary toxic comment classification for English language. The architecture has an input embedding layer connected to 3 convolutional blocks in parallel. Each convolutional block has a fixed filter of width equal to the word vector dimension and filter heights are 3, 4 and 5 for each of the respective parallel blocks. Each convolutional block also contains a max-over-time pooling layer after the convolutional layer. The output of each of the convolutional blocks is then concatenated to a fully connected layer. They used SGD algorithm with mini-batches with a learning rate of 0.005.

4.3.2 BGRU-P

BGRU-P is the best model taken from [44] used for multi-label classification for the detection of toxic comments in English. The architecture takes the input of 200×300 dimensional matrix to the embedding layer followed by a 1D spatial dropout with 40% dropout rate. Two Bidirectional GRU (BGRU) layers are connected to the input embedding layer in parallel. The first GRU layer has 128 units and the second GRU layer has 64 units. The loss function used is focal loss with α=0.25 and γ=5.0. The output of the two BGRU layers is then concatenated by the concatenation layer which is then attached with max pooling and average pooling layers. Both of the pooling layers are further concatenated and attached with a dropout layer with 10% dropout rate, which is then connected to three fully connected layers having 100, 50 and 6 neurons respectively. The last layer in the architecture has sigmoid activation function and its 6 neurons are because of the multi-labeled (6-labeled) classification problem. We fix the last layer to have a single neuron due to the binary nature of the classification problem being addressed in this paper.

4.3.3 CNN+GRU

CNN+GRU is a Convolution-GRU based deep learning neural network taken from [53] that has 100×300 dimensional input embedding layer connected to a dropout layer with dropout rate of 0.2. The following layer is a 1D convolutional layer with RELU as activation function and the number of filters

is 100 where the filter size is 4. A max pool layer with pool size equal to 4 is attached with the convolutional layer, then GRU layer having 100 units is attached which is then attached to a global max pooling layer. Finally, a fully connected layer with the number of classes is attached with softmax as activation function along with elastic-net regularization (L_1 and L_2 norms). They use categorical cross-entropy loss function with Adam optimizer.

4.3.4 Recurrent Neural Networks

The Recurrent Neural Networks used in this study are based on: a) Bidirectional Long-Short Term Memory (BLSTM); b) Bidirectional Gated Recurrent Unit (BGRU). Both of these recurrent architectures are exactly identical except for the recurrent layer, i.e., the recurrent layer in BLSTM has LSTM cells whereas the recurrent layer in BGRU has GRU cells. We describe them considering one model. A sequence of 150×300 dimensional matrix (similar to the CNN) is given as input with 64 comments in each batch to the model. The embedding layer is attached with a 1D spatial dropout layer which is then attached with the recurrent layer. The output from this recurrent layer is then passed on to a feed-forward layer having 50 neurons after a drop out layer and then to the final feed-forward layer having a single neuron with sigmoid activation function.

4.3.5 Tweaked Convolutional Neural Network

We tweak a Convolutional Neural Network (CNN) architecture originally reported in [26]. The first layer in this tweaked architecture is a word-level embedding layer which takes input comments in the form of a real-valued embedding matrix. The dimensions of the embedding matrix are 150×300 where 150 is fixed as the (maximum) length of a comment and 300 is the dimension of the real-valued word vectors. The embedding layer passes the feature maps of the comment embedding matrix to a 1D spatial dropout layer to achieve generalization and reduce over-fitting. The 1D spatial dropout layer drops an entire 1D vector. We attach 5 convolutional blocks in parallel to the 1D spatial dropout layer. Each convolutional block contains a convolutional layer and a global max pooling layer. The convolutional layer in each block has 32 filters (kernels) where the width of the filters is fixed 300 due to the dimension of the word vectors. The heights of the filters are 1,2,3,4,5 in each of the 5 blocks respectively. Hence, the filter dimensions in each of the 5 blocks are 1×300, 2×300, $\ldots$, 5×300 respectively. The global max pooling layer down-samples the entire feature map to a single value, i.e., it takes one maximum value from one entire filter. The output of each of the convolutional blocks is a 32-dimensional feature vector (due to 32 number of filters in each block). We then concatenate the output received from all 5 convolutional blocks via the concatenation layer, producing a 1×160 dimensional vector. Intuitively, to this level, we have combined unigrams, bigrams, trigrams, 4-grams and 5-grams in a 1-dimensional vector. A dropout layer is attached to the concatenation layer

to achieve better generalization up to this level of the network. The learned features in the 1×160 dimensional vector are then hierarchically reduced by using three fully connected layers having 100, 50 and 1 units respectively. We use a single neuron in the last layer due to the binary classification problem addressed in this paper. We train our Convolutional Neural Network by the standard back-propagation algorithm to minimize the above mentioned binary cross-entropy loss function.

This architecture is similar to [26, 18]. The differences are: a) we use 5 convolutional blocks instead of 3 as used in [18]; b) we use 2D convolutions instead of 1D as used in [26]; c) we add 1D spatial drop out layer after the embedding layer; d) we use another dropout layer after concatenating all features produced by the convolutional blocks; e) we hierarchically down-sample the features by introducing 3 fully-connected layers.

5 Experimentation and Results

We used a popular and widely used deep learning framework Keras[4] with Tensorflow backend for the development and training of the deep architectures. For Naïve Bayes, Random Forests and Logistic Regression we used Scikit-learn library whereas for Support Vector Machines we used ThunderSVM[5] because of GPU-support. The simulations have been executed on an NVIDIA 1080 GPU having 8 GB GPU-memory with Ubuntu 16.04 as operating system. The dataset along with all the source codes are available[6].

Each learning model is evaluated using stratified 5-fold cross-validation, i.e., we divide the corpus into five stratified partitions where each divided partition contains almost the same distribution of toxic and non-toxic comments. For each of the five folds, one partition is used as a test set and the remaining partitions are used as the training set. All models are optimized for F1 to ensure a fair comparison. The seed point to split the data into 5 stratified folds is fixed to zero to reproduce the results.

5.1 Evaluation

We report average accuracy, average precision, average recall and average F1 for each model over all folds. Due to the imbalanced nature of the dataset, accuracy can mislead and since F1 is the harmonic mean of precision and recall, we use F1 as the main metric of evaluation in this paper.

[4] `https://keras.io/`

[5] `https://github.com/Xtra-Computing/thundersvm`

[6] `https://github.com/hafizhassaan/Roman-Urdu-Toxic-Comments`

5.2 Hyper-parameter Tuning

We train a total of 40 different learning models depending upon the encoded representation of the comments to detect Roman Urdu toxic comments. Each learning model has its own set of hyper-parameters where each hyper-parameter belongs to its own hyper-parameter space. We explore the hyper-parameters for each model manually by holding one validation set out of the training set in each of the five folds. For logistic regression we tune penalty, C and solver; parameters for support vector machines are penalty, C, kernel, gamma; for random forests, number of estimators, depth of trees, criterion; and the parameters for each deep model include batch size, learning rate, learning rate decay, weight initialization, drop out rate, regularization and early stopping. All hyper-parameters are tuned to optimize the F1 scores on the validation set.

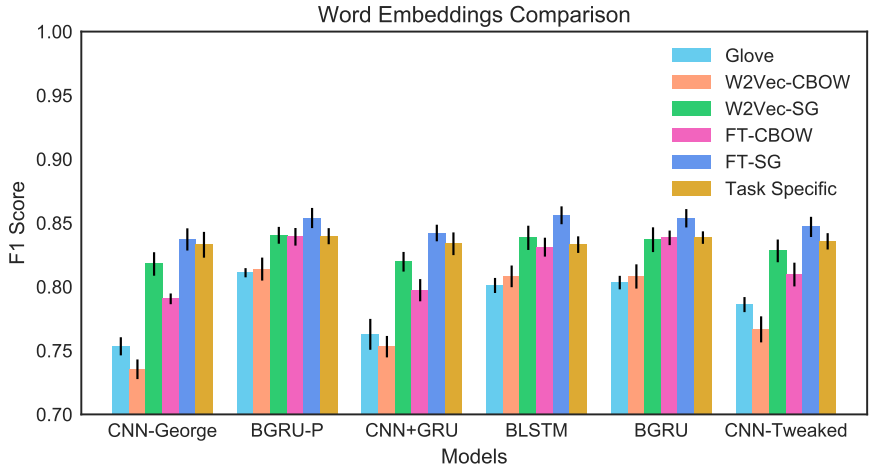

Fig. 1 Comparison of Word Embeddings used with all deep models. The figure is best viewed in color.

5.3 Word Embeddings

We generate Glove, Word2Vec, FastText and task-specific word embeddings for Roman Urdu on our prepared corpus. For Word2Vec and FastText, we train continuous bag-of-words (CBOW) and skip-gram (SG) models separately whereas the task-specific word embeddings are learned as part of training individual models where initialization of the task-specific word vectors is taken as a hyper-parameter in this study. Context window in each of Glove, Word2Vec and FastText is of size 15 and vector dimensions are of size 300. Glove model is trained for 150 epochs whereas Word2Vec and FastText (with both CBOW

and SG models) are trained for 30 epochs. FastText embeddings were based on character 5-grams. We used Gensim[7] library to train Word2Vec models and used open source codes available for training Glove[8] and FastText[9] models.

Each deep model used in this study is trained with each of the generated word embeddings. We show the comparison of the word embeddings used in Figure 1 where the scores are based on the F1 score. It can be seen from Figure 1 that: a) Glove and Word2Vec-CBOW based models give lower F1 scores even than the logistic regression and support vector machines (Table 4) which use simple TF.IDF weights; b) FastText-CBOW based models give better F1 scores than Glove and Word2Vec-CBOW models; c) skip-gram models (for both Word2Vec and FastText) give better F1 scores than CBOW models and the Glove model; d) task-specific word embeddings give better F1 scores as compared to Glove and Word2Vec-CBOW models; e) overall, models based on word embeddings learned with skip-gram model of FastText (FastText-SG) achieve the highest F1 scores. Respective accuracy, precision and recall scores are shown in Table 4.

5.4 Performance of Individual Models

Table 4 shows the results of all experiments performed over the prepared RUT corpus. We show accuracy, precision, recall and F1 separately. Among individual models, Bidirectional LSTM (BLSTM) with skip-gram based FastText word embeddings achieves the highest accuracy score of 95.05% and the highest F1 score of 85.60%. The highest precision (90.77%) is achieved by Bidirectional GRU (BGRU) with skip-gram based FastText word embeddings where its recall is slightly lower than the highest F1 achieving BLSTM. The highest recall among individual models is achieved by SVM which is 83.56%. Comparing the performance of individual deep models while fixing the best learned word embeddings, i.e., skip-gram model of FastText, it can be seen from Table 4 that recurrent architectures (BGRU-P, BLSTM and BGRU) give slightly better F1 scores as compared to the architectures having convolutional layers (CNN-George, CNN+GRU, CNN-Tweaked) over the prepared RUT corpus.

In addition to the scores of individual experiments, we analyze the agreement existing among the learning models in predicting the labels of the comments. For this analysis, we select classical machine learning methods and the best deep models (i.e., deep models with skip-gram based FastText word embeddings). Figure 2 shows the pairwise agreement calculated using Cohen's kappa statistic. Bidirectional GRU (BGRU) and Bidirectional LSTM (BLSTM) have the highest agreement (0.93) in labeling the comments whereas (SVM and LR), (BLSTM and BGRU-P) and (BGRU and BGRU-P) have the next-highest agreement. Further analysis of these pairwise agreements shows that models except Naïve Bayes (NB) and Random Forests (RF) have a strong

[7] https://github.com/RaRe-Technologies/gensim

[8] https://nlp.stanford.edu/projects/Glove/

[9] https://github.com/facebookresearch/FastText/

Table 4 Comparison (in percentages) of all learning models (average of 5 folds).

Models	**Accuracy**	**Precision**	**Recall**	**F1**
NB	91.86 ± 0.27	79.41 ± 1.61	74.09 ± 2.06	76.62 ± 0.85
RF	91.78 ± 0.77	78.92 ± 5.49	75.16 ± 3.54	76.73 ± 0.97
LR	93.67 ± 0.18	83.98 ± 1.75	80.19 ± 1.91	82.00 ± 0.52
SVM	93.87 ± 0.21	82.69 ± 2.00	83.56 ± 1.70	83.08 ± 0.30
Task-specific				
CNN-George	94.25 ± 0.36	87.33 ± 1.44	79.63 ± 1.11	83.29 ± 1.01
BGRU-P	94.45 ± 0.18	87.52 ± 0.34	80.70 ± 1.19	83.97 ± 0.63
CNN+GRU	94.34 ± 0.23	88.53 ± 1.34	78.84 ± 2.18	83.37 ± 0.89
BLSTM	94.30 ± 0.27	88.20 ± 1.77	78.95 ± 0.88	83.30 ± 0.65
BGRU	94.45 ± 0.18	88.12 ± 1.43	80.02 ± 1.20	83.86 ± 0.49
CNN-Tweaked	94.39 ± 0.20	88.45 ± 0.67	79.19 ± 0.85	83.56 ± 0.64
Glove				
CNN-George	91.83 ± 0.21	82.53 ± 1.98	69.38 ± 2.07	75.34 ± 0.71
BGRU-P	93.58 ± 0.19	86.35 ± 2.36	76.55 ± 1.92	81.10 ± 0.36
CNN+GRU	92.15 ± 0.23	83.66 ± 2.40	70.26 ± 3.32	76.27 ± 1.21
BLSTM	93.39 ± 0.13	87.44 ± 0.96	73.93 ± 1.45	80.10 ± 0.59
BGRU	93.40 ± 0.13	86.74 ± 2.25	74.91 ± 2.30	80.33 ± 0.53
CNN-Tweaked	92.73 ± 0.18	83.63 ± 1.93	74.23 ± 2.02	78.61 ± 0.59
Word2Vec CBOW				
CNN-George	90.86 ± 0.22	76.80 ± 0.98	70.56 ± 1.47	73.54 ± 0.77
BGRU-P	93.58 ± 0.22	85.06 ± 0.90	78.06 ± 2.10	81.39 ± 0.90
CNN+GRU	91.65 ± 0.24	80.52 ± 1.79	70.81 ± 2.16	75.31 ± 0.84
BLSTM	93.42 ± 0.23	84.99 ± 1.05	77.08 ± 1.82	80.82 ± 0.85
BGRU	93.44 ± 0.31	85.40 ± 2.19	76.78 ± 2.25	80.81 ± 0.95
CNN-Tweaked	92.17 ± 0.33	82.68 ± 1.19	71.47 ± 1.22	76.66 ± 1.02
Word2Vec Skipgram				
CNN-George	93.75 ± 0.28	86.01 ± 1.19	78.00 ± 1.64	81.79 ± 0.92
BGRU-P	94.50 ± 0.23	87.95 ± 1.55	80.49 ± 1.45	84.04 ± 0.66
CNN+GRU	93.73 ± 0.28	85.08 ± 1.60	79.10 ± 1.40	81.96 ± 0.77
BLSTM	94.46 ± 0.28	88.24 ± 0.53	79.86 ± 1.54	83.84 ± 0.95
BGRU	94.39 ± 0.29	87.75 ± 0.92	80.01 ± 1.66	83.69 ± 0.97
CNN-Tweaked	94.08 ± 0.34	86.70 ± 1.63	79.28 ± 0.69	82.81 ± 0.89
FastText CBOW				
CNN-George	92.72 ± 0.21	82.06 ± 1.67	76.31 ± 1.40	79.06 ± 0.42
BGRU-P	94.57 ± 0.22	89.90 ± 1.60	78.73 ± 1.63	83.92 ± 0.69
CNN+GRU	93.02 ± 0.39	83.59 ± 2.44	76.28 ± 1.32	79.73 ± 0.87
BLSTM	94.23 ± 0.29	87.86 ± 1.97	78.89 ± 1.27	83.11 ± 0.74
BGRU	94.47 ± 0.19	88.43 ± 0.93	79.70 ± 0.92	83.84 ± 0.57
CNN-Tweaked	93.48 ± 0.25	85.28 ± 1.53	77.12 ± 2.24	80.96 ± 0.93
FastText Skipgram				
CNN-George	94.41 ± 0.24	88.05 ± 0.81	79.81 ± 1.79	83.71 ± 0.87
BGRU-P	95.01 ± 0.24	90.17 ± 1.10	81.12 ± 1.52	85.39 ± 0.79
CNN+GRU	94.61 ± 0.21	88.98 ± 1.24	79.97 ± 1.40	84.21 ± 0.65
BLSTM	95.05 ± 0.21	89.78 ± 0.35	81.80 ± 1.20	85.60 ± 0.70
BGRU	95.03 ± 0.25	90.77 ± 1.22	80.59 ± 0.82	85.37 ± 0.72
CNN-Tweaked	94.76 ± 0.19	89.32 ± 0.69	80.56 ± 1.91	84.69 ± 0.79
Ensemble				
All ML (MV)	93.86 ± 0.17	82.51 ± 1.57	**83.68 ± 1.48**	83.06 ± 0.32
All ML (AP)	94.29 ± 0.11	88.84 ± 0.51	78.11 ± 0.35	83.13 ± 0.32
All Deep (MV)	95.19 ± 0.23	92.41 ± 0.38	79.80 ± 1.28	85.64 ± 0.78
All Deep (AP)	95.07 ± 0.26	93.19 ± 0.49	78.35 ± 1.23	85.12 ± 0.88
ML+B. Deep (MV)	**95.30 ± 0.20**	90.41 ± 0.32	82.64 ± 1.13	**86.35 ± 0.66**
ML+B. Deep (AP)	95.28 ± 0.28	92.47 ± 0.37	80.34 ± 1.43	85.97 ± 0.93
B. Deep (MV)	95.16 ± 0.21	90.21 ± 0.49	82.01 ± 1.25	85.91 ± 0.71
B. Deep (AP)	95.19 ± 0.19	92.12 ± 0.25	80.17 ± 1.28	85.72 ± 0.69
All Models (MV)	95.20 ± 0.25	92.37 ± 0.37	79.93 ± 1.42	85.69 ± 0.86
All Models (AP)	95.10 ± 0.24	**93.28 ± 0.47**	78.42 ± 1.05	85.20 ± 0.79

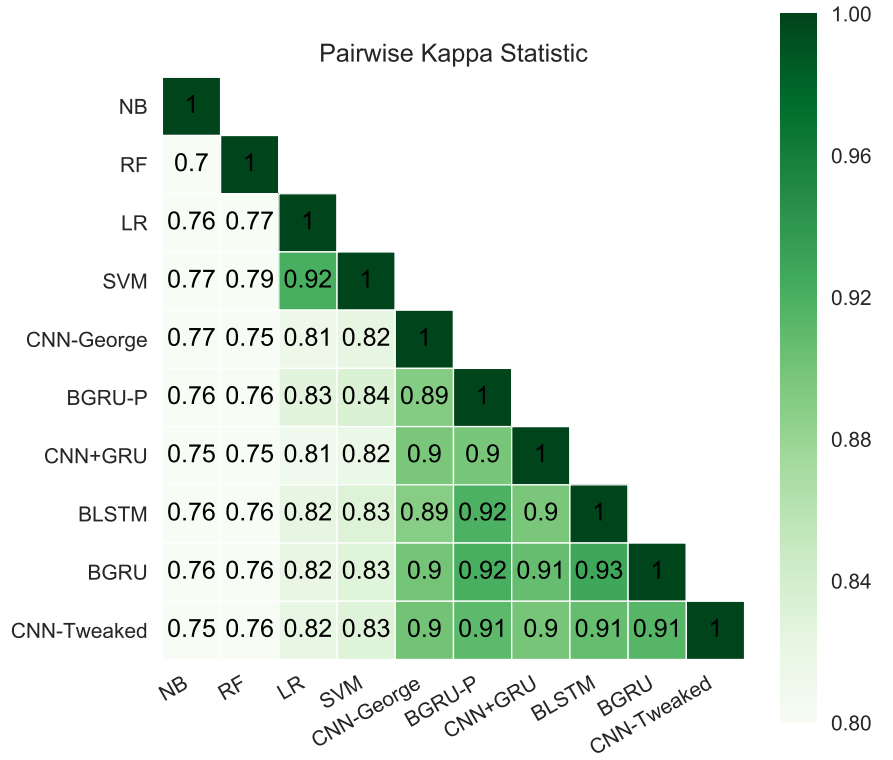

Fig. 2 Pairwise agreement between predictions of ML and Best Deep models in terms of kappa. The figure is best viewed in color.

agreement in label predictions but a proportion of disagreement also exists there. Hence, based on the disagreement among label predictions, the F1 scores can further be improved using ensemble methods, either by taking a majority vote or by taking an average of the predicted probabilities.

5.5 Ensemble Approach

Based on the predictions of individual classifiers, we now report the scores achieved by using ensemble methods over RUT corpus. Overall, we develop five sets of ensembles: 1) ensemble of machine learning models (All ML); 2) ensemble of deep models (All Deep); 3) ensemble of machine learning and the best deep learning models (ML+B. Deep); 4) ensemble of the best deep learning models (B. Deep); 5) ensemble of all models (All Models). We assess these five sets by both a majority vote (MV) and an average probability (AP). Table 4 shows that the highest precision is achieved by the ensemble of all models with average probabilities (All Models (AP)) and the highest recall is achieved by the ensemble of machine learning models with majority vote (All

ML (MV)). The set of the ensemble of machine learning and the best deep learning models with majority voting scheme (ML+B. Deep (MV)) achieve the highest accuracy of 95.30% and the highest F1 of 86.35%.

6 Discussion

The proposed ensemble eloquently classifies the RUT Corpus with an average accuracy of 95.30% and an average F1 score of 86.35%. The confusion matrix on the whole corpus is shown in Table 5. The ratios of misclassification are $(2273/13097) \times 100 = 17.35\%$ for toxic class and $(1148/59674) \times 100 = 1.92\%$ for non-toxic class. By these ratios, we can safely infer that the model makes more mistakes on the toxic class as compared to the non-toxic class. One obvious reason for this relative difference is the class imbalance, i.e., 82% (approx.) of the comments are non-toxic in the RUT Corpus. Another reason could be the class-specific vocabularies which we analyze using I2C and U2C ratios used in [53].

Table 5 Confusion matrix of the best ensemble set

Confusion Matrix		**Predicted**	
		Toxic	Non-Toxic
Actual	Toxic	10824	2273
	Non-Toxic	1148	58526

I2U (Instance to Unique words) ratio of a class divides the number of instances (comments) of that class by the number of unique tokens in that class. This ratio measures the number of comments that may share at least one word. Higher the I2U ratio, the easier it will be to classify the class. As Table 6 shows, the I2U ratio of the non-toxic class is higher as compared to the toxic class, leading our models to perform relatively poor on the toxic class.

U2C (Unique words to Class) ratio of a class divides the number of unique words found only in that class excluding the words present in the other class by the total number of unique words existing in that class. We can see from the Table 6, U2C ratio for the toxic class is 0.49 and for the non-toxic is 0.83, this means that toxic class has 49% of words unique to itself only while the remaining 51% overlap with the non-toxic class, and, the non-toxic class has 83% of words unique to itself only while the remaining 17% overlaps with the toxic class, therefore, more than half of the unique words in the toxic class are overlapped with the non-toxic class which leads our models to make relatively more errors in identification of the toxic class.

Both I2U and U2C ratios make the non-toxic class easier to predict as compared to the toxic class, which explains the significantly higher misclassification ratio in the toxic class. Moreover, by looking at I2U and U2C ratios we can also say RUT Corpus has a diverse set of Roman Urdu vocabulary.

A few examples of correctly predicted comments (true positives and true negatives) can be seen in Table 7. Comment # 2 shown in the table lies in the

Table 6 RUT Corpus specific vocabulary details

Class	Unique Tokens	Class-Specific Unique Tokens	I2U	U2C
Toxic	25366	12470	0.5163	0.4916
Non-Toxic	78775	65879	0.7575	0.8362

set of true positives. The reason we highlight comment # 2 is that it contains two toxic words with asterisks. The asterisks cover the true identity (spelling) of the word and make it hard for a classifier to identify its true class. The two words are "c**ot" and "Gaan*u" which actually are "choot" meaning [pussy] and "Gaandu" meaning [asshole]. The comments 7-12 in Table 7 are some examples of misclassified comments (false positives and false negatives). Comment # 8 is an interrogative comment, actually having a non-toxic tone, but the model is predicting it as toxic because of the word "Jahil" meaning [illiterate]. Further investigation reveals that the model makes errors on interrogative type of comments if they contain toxic words. The comments 11-12 are examples of false negatives. Both of the comments contain a single toxic word, i.e., "hurami" meaning [bastard] and "kanjeriya" meaning [prostitutes] respectively. The word "hurami" occurs twice in the corpus whereas the word "kanjeriya" occurs only once in the corpus, making these comments hard examples for the classifier. We present 3 more sentences not given in the table: 1) "Chal chuthiye" translated in English as [fuck off, idiot]; 2) "Sab Kanjjarr Log hain yeh" translated as [All these people are pimps]; 3) "Aba bahencho asa film mat bana" translated as [Oh sister fucker do not make this type of films]. These 3 comments are false negatives as well. The reason here is the lack of proper spelling structure in Roman Urdu. So we hypothesize that our model may make errors when it encounters some varied spellings. Furthermore, the model lacks familiarity with the ideograms (symbol representation of words), for example, "(.)(.) chooso" translated as [suck the boobs (symbol)] are mostly misclassified. One certain reason for such type of misclassification is the removal of punctuation marks in the pre-processing phase.

7 Conclusion

This paper presents the first treatment of toxic comment detection in Roman Urdu which is an under-resourced language. Roman Urdu is an informal writing style of Urdu that is used primarily on social media by millions of users with roots in South Asia. The detection of toxic comments in Roman Urdu is challenging due to the scarcity of language resources and the unavailability of appropriate corpora. We introduce a large and first-ever corpus for the study of toxic comments in Roman Urdu which contains over 72 thousand labeled comments with a strong inter-annotator agreement. We use several existing techniques that perform well over English toxic datasets including classical text classification methods and recent deep learning based models. Due to the absence of pre-trained word embeddings in Roman Urdu, we develop several

Table 7 Sample predicted comments of RUT Corpus

	#	Comments	Actual	Predicted
tp	1	teri ami jan ki kus main lun maro [I want to fuck your mum's pussy with cock.]	1	1
	2	Abey teri ben ki c**ot Gaan*u [Oh, your sister's pussy, (you) asshole]	1	1
	3	Chaat merey tatte jaise aaloo ka pakoora [Lick my balls as if they are potato pakoras.]	1	1
tn	4	Galtya har chez me nikal ti han agr koi gor karay to [Mistakes exist in everything if someone observes.]	0	0
	5	Allah pak hum sab ko hidayat naseeb frmye [May Allah bless us with guidance.]	0	0
	6	Bhai me aap ki baat say agree karta ho. [I agree with you brother.]	0	0
fp	7	Galat logo ki izzat karna qom par zolam karna hy [To respect wrong people is to abuse the nation.]	0	1
	8	aap aalim banna chaho ge yaa jahil? [Would you like to become a scholar or an illiterate?]	0	1
	9	Ak bjhy charag ne mera hath jla dia [An quenched lamp burnt my hand]	0	1
fn	10	In b gherton ko jhannum m bhajdo [Send these brazens to hell!]	1	0
	11	Hahahaha tum hurami ho jo jang ki bat karte ho [Hahahaha you are a bastard that you talk about war]	1	0
	12	Ye kanjeriya to pesey ke liye koch be kr skti hn [These prostitutes can do anything for money.]	1	0

word embeddings using Glove, Word2Vec and FastText techniques along with task-specific word vectors learned within the classification task. The results achieved over the prepared corpus show that: a) learning task-specific word vectors performs better than learning Glove and continuous bag-of-words approach of Word2Vec models; b) word embeddings learned using the skip-gram model of FastText achieve the highest empirical scores among all of the learned word embeddings; c) ensembling the individual models enhances the overall scores over RUT corpus, reaching an F1 score of 86.35%. Moreover, this paper amply discusses the performance of ensemble highlighting characteristics that it can and cannot detect reliably.

We have made the labeled RUT corpus available to the research community for future work. We believe this resource will engender better models for Roman Urdu toxic comment detection.

Acknowledgements We thank Louis Bruyns Foundation, Belgium, for their support to complete this research study.

References

1. Agrawal, S., Awekar, A.: Deep learning for detecting cyberbullying across multiple social media platforms. In: Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France,

March 26-29, 2018, Proceedings, pp. 141–153 (2018). DOI 10.1007/978-3-319-76941-7_11
2. van Aken, B., Risch, J., Krestel, R., Löser, A.: Challenges for toxic comment classification: An in-depth error analysis. In: Proceedings of the 2nd Workshop on Abusive Language Online, ALW@EMNLP 2018, Brussels, Belgium, October 31, 2018, pp. 33–42 (2018). URL https://aclanthology.info/papers/W18-5105/w18-5105
3. Al-garadi, M.A., Varathan, K.D., Ravana, S.D.: Cybercrime detection in online communications: The experimental case of cyberbullying detection in the twitter network. Computers in Human Behavior **63**, 433–443 (2016). DOI 10.1016/j.chb.2016.05.051
4. Albadi, N., Kurdi, M., Mishra, S.: Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In: U. Brandes, C. Reddy, A. Tagarelli (eds.) IEEE/ACM 2018 International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2018, Barcelona, Spain, August 28-31, 2018, pp. 69–76. IEEE Computer Society (2018). DOI 10.1109/ASONAM.2018.8508247. URL https://doi.org/10.1109/ASONAM.2018.8508247
5. Ameer, I., Siddiqui, M.H.F., Sidorov, G., Gelbukh, A.F.: CIC at semeval-2019 task 5: Simple yet very efficient approach to hate speech detection, aggressive behavior detection, and target classification in twitter. In: J. May, E. Shutova, A. Herbelot, X. Zhu, M. Apidianaki, S.M. Mohammad (eds.) Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019, pp. 382–386. Association for Computational Linguistics (2019). URL https://aclweb.org/anthology/papers/S/S19/S19-2067/
6. Aulia, N., Budi, I.: Hate speech detection on indonesian long text documents using machine learning approach. In: Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence, ICCAI 2019, Bali, Indonesia, April 19-22, 2019., pp. 164–169. ACM (2019). DOI 10.1145/3330482.3330491. URL https://doi.org/10.1145/3330482.3330491
7. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, Perth, Australia, April 3-7, 2017, pp. 759–760. International World Wide Web Conferences Steering Committee (2017). DOI 10.1145/3041021.3054223
8. Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Pardo, F.M.R., Rosso, P., Sanguinetti, M.: Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In: J. May, E. Shutova, A. Herbelot, X. Zhu, M. Apidianaki, S.M. Mohammad (eds.) Proceedings of the 13th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2019, Minneapolis, MN, USA, June 6-7, 2019, pp. 54–63. Association for Computational Linguistics (2019). URL https://aclweb.org/anthology/papers/S/S19/S19-2007/

9. Bilal, A., Rextin, A., Kakakhel, A., Nasim, M.: Roman-txt: forms and functions of roman urdu texting. In: Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2017, Vienna, Austria, pp. 15:1–15:9. ACM (2017). DOI 10.1145/3098279.3098552
10. Bögel, T.: Urdu - roman transliteration via finite state transducers. In: Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing, FSMNLP 2012, Donostia-San Sebastiían, Spain, July 23-25, 2012, pp. 25–29 (2012). URL http://aclweb.org/anthology/W/W12/W12-6204.pdf
11. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017). URL https://transacl.org/ojs/index.php/tacl/article/view/999
12. Chung, Y., Kuzmenko, E., Tekiroglu, S.S., Guerini, M.: CONAN - counter narratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In: A. Korhonen, D.R. Traum, L. Màrquez (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 2819–2829. Association for Computational Linguistics (2019). URL https://www.aclweb.org/anthology/P19-1271/
13. Eberhard, D.M., Simons, G.F., Fennig, C.D.: Urdu. Ethnologue: Languages of the World Twenty-second edition. Dallas, Texas: SIL International (2019). URL https://www.ethnologue.com/language/urd. "Last accessed: 25-07-2019"
14. ElSherief, M., Nilizadeh, S., Nguyen, D., Vigna, G., Belding, E.M.: Peer to peer hate: Hate speech instigators and their targets. In: Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018., pp. 52–61. AAAI Press (2018). URL https://aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17905
15. Fersini, E., Rosso, P., Anzovino, M.: Overview of the task on automatic misogyny identification at ibereval 2018. In: P. Rosso, J. Gonzalo, R. Martínez, S. Montalvo, J.C. de Albornoz (eds.) Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018., *CEUR Workshop Proceedings*, vol. 2150, pp. 214–228. CEUR-WS.org (2018). URL http://ceur-ws.org/Vol-2150/overview-AMI.pdf
16. Fortuna, P., Bonavita, I., Nunes, S.: Merging datasets for hate speech classification in italian. In: T. Caselli, N. Novielli, V. Patti, P. Rosso (eds.) Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018., *CEUR Workshop*

Proceedings, vol. 2263. CEUR-WS.org (2018). URL http://ceur-ws.org/Vol-2263/paper037.pdf
17. Gambäck, B., Sikdar, U.K.: Using convolutional neural networks to classify hate-speech. In: Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017, pp. 85–90 (2017). URL https://aclanthology.info/papers/W17-3013/w17-3013
18. Georgakopoulos, S.V., Tasoulis, S.K., Vrahatis, A.G., Plagianakos, V.P.: Convolutional neural networks for toxic comment classification. In: Proceedings of the 10th Hellenic Conference on Artificial Intelligence, SETN 2018, Patras, Greece, July 09-12, 2018, pp. 35:1–35:6. ACM (2018). DOI 10.1145/3200947.3208069
19. Ghulam, H., Zeng, F., Li, W., Xiao, Y.: Deep learning-based sentiment analysis for roman urdu text. In: 2018 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2018, Beijing, China, October 19-21, 2018, *Procedia Computer Science*, vol. 147, pp. 131–135. Elsevier (2018). DOI 10.1016/j.procs.2019.01.202
20. Guggilla, C., Miller, T., Gurevych, I.: CNN- and lstm-based claim classification in online user comments. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pp. 2740–2751 (2016). URL http://aclweb.org/anthology/C/C16/C16-1258.pdf
21. Haidar, B., Chamoun, M., Serhrouchni, A.: Multilingual cyberbullying detection system: Detecting cyberbullying in arabic content. In: 1st Cyber Security in Networking Conference, CSNet 2017, Rio de Janeiro, Brazil, October 18-20, 2017, pp. 1–8. IEEE (2017). DOI 10.1109/CSNET.2017.8242005. URL https://doi.org/10.1109/CSNET.2017.8242005
22. Hee, C.V., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., Pauw, G.D., Daelemans, W., Hoste, V.: Detection and fine-grained classification of cyberbullying events. In: G. Angelova, K. Bontcheva, R. Mitkov (eds.) Recent Advances in Natural Language Processing, RANLP 2015, 7-9 September, 2015, Hissar, Bulgaria, pp. 672–680. RANLP 2015 Organising Committee / ACL (2015). URL http://aclweb.org/anthology/R/R15/R15-1086.pdf
23. Hosseinmardi, H., Rafiq, R.I., Han, R., Lv, Q., Mishra, S.: Prediction of cyberbullying incidents in a media-based social network. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016, pp. 186–192 (2016). DOI 10.1109/ASONAM.2016.7752233
24. Ibrahim, M., Torki, M., El-Makky, N.: Imbalanced toxic comments classification using data augmentation and deep learning. In: 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018, pp. 875–878. IEEE (2018). DOI 10.1109/ICMLA.2018.00141

25. Khan, O., Karim, A.: A rule-based model for normalization of SMS text. In: IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7-9, 2012, pp. 634–641 (2012). DOI 10.1109/ICTAI.2012.91. URL https://doi.org/10.1109/ICTAI.2012.91
26. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1746–1751 (2014). URL http://aclweb.org/anthology/D/D14/D14-1181.pdf
27. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., vol. 333, pp. 2267–2273. AAAI Press (2015). URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9745
28. Lee, H.S., Lee, H.R., Park, J.U., Han, Y.S.: An abusive text detection system based on enhanced abusive and non-abusive word lists. Decision Support Systems **113**, 22–31 (2018). DOI 10.1016/j.dss.2018.06.009
29. Lee, S., Kim, H.: Why people post benevolent and malicious comments online. Communications of the ACM **58**(11), 74–79 (2015). DOI 10.1145/2739042. URL https://doi.org/10.1145/2739042
30. Malmasi, S., Zampieri, M.: Challenges in discriminating profanity from hate speech. Journal of Experimental & Theoretical Artificial Intelligence **30**(2), 187–202 (2018). DOI 10.1080/0952813X.2017.1409284. URL https://doi.org/10.1080/0952813X.2017.1409284
31. Mehmood, K., Essam, D., Shafi, K.: Sentiment analysis system for roman urdu. In: Science and Information Conference, pp. 29–42. Springer (2018)
32. Mehmood, K., Essam, D., Shafi, K., Malik, M.K.: Discriminative feature spamming technique for roman urdu sentiment analysis. IEEE Access **7**, 47,991–48,002 (2019). DOI 10.1109/ACCESS.2019.2908420. URL https://doi.org/10.1109/ACCESS.2019.2908420
33. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pp. 3111–3119 (2013). URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality
34. Nobata, C., Tetreault, J.R., Thomas, A., Mehdad, Y., Chang, Y.: Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016, pp. 145–153 (2016). DOI 10.1145/2872427.2883062. URL https://doi.org/10.1145/2872427.2883062
35. Obadimu, A., Mead, E., Hussain, M.N., Agarwal, N.: Identifying toxicity within youtube video comment. In: R. Thomson, H. Bisgin, C.L.

Dancy, A. Hyder (eds.) Social, Cultural, and Behavioral Modeling - 12th International Conference, SBP-BRiMS 2019, Washington, DC, USA, July 9-12, 2019, Proceedings, *Lecture Notes in Computer Science*, vol. 11549, pp. 214–223. Springer (2019). DOI 10.1007/978-3-030-21741-9_22. URL https://doi.org/10.1007/978-3-030-21741-9_22

36. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1532–1543 (2014). URL http://aclweb.org/anthology/D/D14/D14-1162.pdf
37. Poletto, F., Stranisci, M., Sanguinetti, M., Patti, V., Bosco, C.: Hate speech annotation: Analysis of an italian twitter corpus. In: R. Basili, M. Nissim, G. Satta (eds.) Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017), Rome, Italy, December 11-13, 2017., *CEUR Workshop Proceedings*, vol. 2006. CEUR-WS.org (2017). URL http://ceur-ws.org/Vol-2006/paper024.pdf
38. Ptaszynski, M., Eronen, J.K.K., Masui, F.: Learning deep on cyberbullying is always better than brute force. In: IJCAI 2017 3rd Workshop on Linguistic and Cognitive Approaches to Dialogue Agents (LaCATODA 2017), Melbourne, Australia, August, pp. 19–25 (2017)
39. Rafae, A., Qayyum, A., Moeenuddin, M., Karim, A., Sajjad, H., Kamiran, F.: An unsupervised method for discovering lexical variations in roman urdu informal text. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pp. 823–828 (2015). URL http://aclweb.org/anthology/D/D15/D15-1097.pdf
40. Reynolds, K., Kontostathis, A., Edwards, L.: Using machine learning to detect cyberbullying. In: 10th International Conference on Machine Learning and Applications and Workshops, ICMLA 2011, Honolulu, Hawaii, USA, December 18-21, 2011. Volume 2: Special Sessions and Workshop, pp. 241–244 (2011). DOI 10.1109/ICMLA.2011.152. URL https://doi.org/10.1109/ICMLA.2011.152
41. Risch, J., Krestel, R.: Aggression identification using deep learning and data augmentation. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 150–158 (2018)
42. Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., Wojatzki, M.: Measuring the reliability of hate speech annotations: The case of the european refugee crisis. CoRR **abs/1701.08118** (2017). URL http://arxiv.org/abs/1701.08118
43. Rybinski, M., Miller, W., Ser, J.D., Bilbao, M.N., Montes, J.F.A.: On the design and tuning of machine learning models for language toxicity classification in online platforms. In: J.D. Ser, E. Osaba, M.N. Bilbao, J.J.S. Medina, M. Vecchio, X. Yang (eds.) Intelligent Distributed Computing XII, 12th International Symposium on Intelligent Distributed Computing, IDC 2018, Bilbao, Spain, 15-17 October 2018, *Studies in Computational Intel-*

ligence, vol. 798, pp. 329–343. Springer (2018). DOI 10.1007/978-3-319-99626-4_29. URL https://doi.org/10.1007/978-3-319-99626-4_29

44. Saeed, H.H., Shahzad, K., Kamiran, F.: Overlapping toxic sentiment classification using deep neural architectures. In: 2018 IEEE International Conference on Data Mining Workshops, ICDM Workshops, Singapore, Singapore, November 17-20, 2018, pp. 1361–1366. IEEE (2018). DOI 10.1109/ICDMW.2018.00193. URL https://doi.org/10.1109/ICDMW.2018.00193
45. Sanguinetti, M., Poletto, F., Bosco, C., Patti, V., Stranisci, M.: An italian twitter corpus of hate speech against immigrants. In: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (eds.) Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA) (2018). URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/710.html
46. Santosh, T., Aravind, K.: Hate speech detection in hindi-english code-mixed social media text. In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, COMAD/CODS 2019, Kolkata, India, January 3-5, 2019, pp. 310–313. ACM (2019). DOI 10.1145/3297001.3297048. URL https://doi.org/10.1145/3297001.3297048
47. Shtovba, S., Shtovba, O., Petrychko, M.: Detection of social network toxic comments with usage of syntactic dependencies in the sentences. In: Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019), Zaporizhzhia, Ukraine, April 15-19, 2019., *CEUR Workshop Proceedings*, vol. 2353, pp. 313–323. CEUR-WS.org (2019). URL http://ceur-ws.org/Vol-2353/paper25.pdf
48. Sood, S.O., Antin, J., Churchill, E.F.: Profanity use in online communities. In: CHI Conference on Human Factors in Computing Systems, CHI '12, Austin, TX, USA - May 05 - 10, 2012, pp. 1481–1490 (2012). DOI 10.1145/2207676.2208610. URL https://doi.org/10.1145/2207676.2208610
49. Sutejo, T.L., Lestari, D.P.: Indonesia hate speech detection using deep learning. In: M. Dong, M.A. Bijaksana, H. Sujaini, A. Romadhony, F.Z. Ruskanda, E. Nurfadhilah, L.R. Aini (eds.) 2018 International Conference on Asian Language Processing, IALP 2018, Bandung, Indonesia, November 15-17, 2018, pp. 39–43. IEEE (2018). DOI 10.1109/IALP.2018.8629154. URL https://doi.org/10.1109/IALP.2018.8629154
50. Vigna, F.D., Cimino, A., Dell'Orletta, F., Petrocchi, M., Tesconi, M.: Hate me, hate me not: Hate speech detection on facebook. In: A. Armando, R. Baldoni, R. Focardi (eds.) Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy, January 17-20, 2017., *CEUR Workshop Proceedings*, vol. 1816, pp. 86–95. CEUR-WS.org (2017). URL http://ceur-ws.org/Vol-1816/paper-09.pdf
51. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pp. 88–93. The Association for Computational Linguistics (2016). URL http://aclweb.org/anthology/N/N16/N16-2013.pdf

52. Watanabe, H., Bouazizi, M., Ohtsuki, T.: Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access **6**, 13,825–13,835 (2018). DOI 10.1109/ACCESS.2018.2806394. URL https://doi.org/10.1109/ACCESS.2018.2806394
53. Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on twitter using a convolution-gru based deep neural network. In: European Semantic Web Conference - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, pp. 745–760. Springer (2018)
54. Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., Xu, B.: Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pp. 3485–3495 (2016). URL http://aclweb.org/anthology/C/C16/C16-1329.pdf