Expert Systems With Applications 215 (2023) 119342

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments

Hiren Madhu [a,c], Shrey Satapara [b,c,f], Sandip Modha [c,*], Thomas Mandl [d,b], Prasenjit Majumder [b,e]

[a] Indian Institute of Science, Bangalore, India
[b] DAIICT, Gandhinagar, Gujarat, India
[c] LDRP-ITR, Gandhinagar, Gujarat, India
[d] University of Hildesheim, Hildesheim, Germany
[e] TCG CREST, Kolkata, India
[f] Indian Institute of Technology, Hyderabad, India

ARTICLE INFO

Keywords:
Hate Speech
Natural Language Processing
Evaluation
Conversational Analysis
Benchmark
Transformer

ABSTRACT

The spread of Hate Speech on online platforms is a severe issue for societies and requires the identification of offensive content by platforms. Research has modeled Hate Speech recognition as a text classification problem that predicts the class of a message based on the text of the message only. However, context plays a huge role in communication. In particular, for short messages, the text of the preceding tweets can completely change the interpretation of a message within a discourse. This work extends previous efforts to classify Hate Speech by considering the current and previous tweets jointly. In particular, we introduce a clearly defined way of extracting context. We present the development of the first dataset for conversational-based Hate Speech classification with an approach for collecting context from long conversations for code-mixed Hindi (ICHCL dataset). Overall, our benchmark experiments show that the inclusion of context can improve classification performance over a baseline. Furthermore, we develop a novel processing pipeline for processing the context. The best-performing pipeline uses a fine-tuned SentBERT paired with an LSTM as a classifier. This pipeline achieves a macro F1 score of 0.892 on the ICHCL test dataset. Another KNN, SentBERT, and ABC weighting-based pipeline yields an F1 Macro of 0.807, which gives the best results among traditional classifiers. So even a KNN model gives better results with an optimized BERT than a vanilla BERT model.

1. Introduction

The increase of the availability and affordability of state-of-the-art Internet mobile data technologies in particular in developing countries has led to a multifold growth in the social media user base. This huge user base fueled a substantial increase in Hate Speech posts on social media platforms. These platforms allow all members to express their opinions and perspectives irrespective of age, level of expertise or any other feature. This great opportunity to get heard online has also generated many issues. The lack of restrictions and the freedom given to users motivates many to use defamatory language to tarnish the reputation of other users or to outright threaten them. The spread of Hate Speech on online platforms is a serious issue for societies. In order to maintain a rationale discourse which allows all voices to participate without being threatened requires action against offensive and hateful statements. However, in many developing countries, the regulation of online communication needs to be improved by introducing appropriate laws and rules regarding the use of AI for detecting problematic content (Brown, 2020). As a first step, platforms need to identify problematic content. Due to the sheer number of online messages, this can only be done with the help of AI based text classification tools. In addition to the enormous data volume, further difficulties arise from the multilingual nature of the content which is often written in code-mixed script. Furthermore, the conversational dialogue-style of the threads of messages and its multi-modal features obstruct platforms to capture and either moderate or delete Hate Speech posts. Because messages on social media are short, the interpretation of the content requires knowledge about the previous conversation as well as the underlying discourse. Systems need to consider this contexts in order to be successful.

* Corresponding author.
E-mail addresses: hirenmadhu16@gmail.com (H. Madhu), shreysatapara@gmail.com (S. Satapara), sjmodha@gmail.com (S. Modha), mandl@uni-hildesheim.de (T. Mandl), prasenjit.majumder@gmail.com (P. Majumder).

https://doi.org/10.1016/j.eswa.2022.119342
Received 23 September 2021; Received in revised form 29 May 2022; Accepted 21 November 2022
Available online 25 November 2022

1.1. Perspectives from other disciplines

Hate is a typical type of human behavior that involves an intense feeling of dislike for someone's behavior, class, disability, ethnicity, gender, physical appearance, race, religion, and sexual orientation. In a broad sense, we can capture hate as defined above. An impression of hate is commonly available on social media like Reddit, Facebook, Twitter, etc. There are several studies (Judge & Nel, 2018; Saha et al., 2019) that identify the objective relationship between the hater and the hated in a causality framework. Classification and identification of speech intended to degrade, intimidate, or incite violence or prejudicial action against someone is an important study and need of the hour. Hate speech online has led to a global increase in violence towards minorities, including mass shootings, and other violent acts.

Hate Speech has been researched from a linguistic perspective in order to find qualitative types of hate. Political scientist and scholars from Law observe and discuss the regulations of governments against hate speech and in the conflict between censorship and freedom of expression. Sociological research has analyzed the reasons for the emergence of hate speech and the social conditions for unsocial online behavior. AI has focused on the recognition of Hate Speech from a supervised machine learning point of view. This direction is further elaborated in the state of the art section below.

1.2. Defining context in conversational dialogue

In a conversation, two or more agents are participating. If there is a difference of opinion or difference in the belief system among them, there is a chance that feelings of dislike start to evolve. It may eventually end up as a personal attack. In many countries, like India, personal attacks in cyberspace are considered a cognitive offense. India is a multilingual country where many scripts are used. There are 23 scheduled languages and 12 scripts used in digital media. Social media informal texts generated by the Indian diaspora are multi-scripted (code-switch) and multilingual (“jab we met”). Navigating through this huge amount of multi-script and multilingual text and identifying offensive hate speech is a non-trivial problem. Several cultural contexts need to be analyzed automatically in an ideal situation.

This paper aims to study the various forms of problematic content such as aggressiveness, hate, offensive, abusive content in standalone (without context) as well as in conversational dialogue (with context) on online platforms. We attempt to explore this problem by first constructing a benchmark human-annotated dataset (ICHCL dataset). Next, we set up some experiments on ICHCL dataset. To the best of our knowledge, this is the first dataset of its kind.

1.3. Conversational hate speech identification problem formulation

Hate Speech identification needs to address the issue of the context of the social media post. Due to the fact that messages on social media are often short and relate to or react to other. The additional knowledge about the conversion into which a tweet or posting is embedded can improve the classification performance. The problem which this research addresses is the identification of problematic content in social media conversational dialogue. Fig. 1 present the problem we are trying to solve in this paper. The screenshot from Twitter illustrates the problem. The parent tweet which was posted at 2:30 am on May 11th is expressing hate towards Muslim countries with profanity, regarding the controversy happening during the recent Israel-Palestine conflict. The two comments on the tweet say “Amen” which means “truly” or “it is true”, and “let it be so” in Persian. These two comments exhibit their hateful and offensive character only when the reader is aware of the parent tweet to which they refer and which they confirm. If the two tweet would be presented in without this context to an annotator, they would not be classified as Hate Speech or offensive content. But if the reader considers the context of the conversation it becomes obvious that these comments are supporting the hate expressed in the parent tweet. Consequently, these comments need to be considered as Hate Speech as well. Currently, almost all Hate Speech datasets are created at the tweet level and they ignore the dialogue within threads. It is necessary to assemble datasets which allow systems to optimize classification by exploiting the structure and timeline of the threads. By doing that, the classification can employ context information.

Another example representing conversational hate with Hindi–English code mixed text is following and is displayed in Fig. 2:

- **The Source Tweet:** Modi Ji COVID situation ko solve karne ke liye ideas maang rahe the. Mera idea hai resignation dedo please...
- **Translation:** Modi ji (PM of India) was asking for ideas to solve the covid situation of India. My idea to him is to resign.
- **The Comment:** Doctors aur Scientists se manga hai. Chutiyo se nahi. Baith niche. [Profane]
- **Translation:** They have asked Doctors and Scientists. Not fuckers. Sit down. [Profane]
- **The reply:** You totally nailed it, can't stop laughing. [Hate]

The comment is expressing obvious hate towards the author of the primary tweet. The reply has a positive sentiment. Reading only the reply, no user would perceive the message as hateful or offensive. But when the context of the comment is considered as well, it becomes obvious that the positive sentiment is actually positive in favor of the hate expressed in the first message. This shows that the reply is actually supporting the hate expressed towards the author of the source tweet in the comment. And hence, the reply itself is also hateful content. Any Hate Speech research needs to address this issue. We propose a dataset and analysis using various embeddings and Machine Learning algorithms for the problem.

1.4. Context research

Context is a vague concept that is used in many domains in Computer Science. It is most often associated with the use of additional information from resources from related sources. Within NLP, context is often an important topic. For example, LSTMs are a method using context from previous word to understand the current word in a sequence. In Information Retrieval, context is often used to include textual resources from other interactions of the user to obtain a better description of interests (Mandl & Womser-Hacker, 2005). Within Cloud Computing, the problem of task failure prediction has been framed as a context research problem. One approach (Bala & Chana, 2015) used machine learning classifiers such as logistic regression, SVM, and Naive Bayes to predict task failure using resource utilization parameters such as CPU utilization, RAM, Disk Storage and Bandwidth utilization as input . The ORCON model used context histories from Foursquare and Twitter check-ins for the context prediction. Furthermore, the model also supports the best prediction algorithm according to the situation. The model also addresses important aspects of ubiquitous computing such as formal context representation and privacy (da Rosa et al., 2016). The MultCComp model is a multi-temporal context-aware system for competences management. The model used workers' past and present contexts to assist them to develop their competence (Rosa et al., 2015). Context is also used for project management. A computational model which predicts potential risk assists stakeholders control these risks at different points in the life cycle of projects. The model used context histories to predict the potential risks in the new projects. The authors used databases of 153 projects as context histories and achieved 84 % accuracy on ongoing projects. The authors concluded that context histories help managers to make assertive project planning by exploiting risk recommendations by the system (Filippetto et al., 2021). The CHSPAM model discovers sequential patterns in context histories databases and monitors the evolution of these patterns over time (Dupont et al., 2020). For social media analysis, the context of temporal patterns will be an important future research topic where many of these ideas can be applied.

Fig. 1. Hate and Profane Conversational Dialogue Example.

Fig. 2. Code-mixed Profane Conversational Dialogue Example.

1.5. Contribution and overview

The analysis of context within a conversation on social media is necessary for understanding messages fully. However, most datasets for benchmarking Hate Speech are still designed without context and provide only the text to be classified. Only limited work for analyzing context for Hate Speech detection has been done. As Section 2.3 will elaborate, datasets available either provide only a very limited amount

of context (Pavlopoulos et al., 2020) or are built upon an existing dataset without context and augmented which does not capture context for all messages (Menini et al., 2020). The dataset presented here was designed and collected with the goals to investigate the context dependency of Hate Speech and the goal to include long contexts (see Section 3). This ICHCL data set is also the first Hate Speech dataset including context for a language other than only English. It is assembled by extracting messages from the social media platform Twitter. We developed the first formalized and standardized format for presenting context and created specific extraction tools for threads in order to implement this task. This format is based on topics and requires a dedicated annotation method. Given this dataset, we developed a model for processing context optimally and achieves some 10% above the state of the art as obtained in a shared task. Our classification experiments apply diverse traditional and state of the art text classification systems based on deep learning representations (Modha et al., 2022). Our method assigns different weights to tweets, comments and replies when combining them. The best weights for this combination have been obtained by extensive experiments.

In short, these are the research objectives of this paper.

1. To develop a hate and offensive speech dataset from the social media conversational dialogue.
2. To capture the context of social media communication to improve the performance of the offensive text classification.
3. To represent conversational dialogue in a fixed-length vector for transformer models.
4. To experiment with a variety of classifiers to obtain the best performance on the proposed ICHCL dataset.

The rest of the paper is organized as follows. Section 2 reviews related work. In Section 3, we present the procedure of collection of the data, the structure of the ICHCL dataset and its annotation. Section 4 reports the experiments with diverse machine learning algorithms In Section 5, we introduce a novel technique to process contextual data with deep neural networks. Section 6 sums up our work and gives ideas for future research.

2. Related work

In this section, we review the literature on hate speech, offensive language from various perspectives, such as formulation of problem, detection, code-mixed text and other text classification problems that make use of the conversational structure of the data and use a context model. We also shed light on some of the initial approaches used to filter the conversational dialogue presented in the ICHCL dataset at HASOC forum (Modha et al., 2021b)

2.1. Hate speech datasets and challenges

The problem of hate speech has led to the creation of many datasets. They often model the challenge differently. Consequently, there are notions like toxicity (Pavlopoulos et al., 2020), aggressiveness (Aroyehun & Gelbukh, 2018) or offensiveness (Davidson et al., 2017). Some datasets were created specifically for issues like hate against migrants, have against other ethnic groups (Pronoza et al., 2021) and hate against women (Parikh et al., 2021). TRAC-1 (Kumar et al., 2018), HateEval (Basile et al., 2019), and OLID (Zampieri et al., 2019) are popular datasets in the research community that look at each post individually without providing any context to determine the label of the posts.

The work reported in this article builds upon experiences from the HASOC shared task[1] and expands the definition adopted there. The shared task HASOC (Mandl et al., 2020) created a large multilingual dataset for hate Speech identification. HASOC was introduced in 2019. The first HASOC track focused on the identification of Hate Speech in Indo-European languages namely Hindi, English and German (Mandl et al., 2019). In 2020, there were 2 sub-tracks within HASOC, the second subtrack was for Tamil and Malayalam in native and Latin script. The first track contained 2 subtasks for the three languages Hindi, English and German. Subtask A introduced a binary classification into problematic content and other content. Subtask B was a fine-grained classification, further classifying problematic content between Hate, Offensive and profane content. A system submitted by Madhu et al. (2020) proposed an approach suggested by Kim (2014a) which adopts a CNN for text classification. Authors also provided an in-depth comparison of models including SVM, Naive Bayes, KNN, Deep Neural Network, CNNs with various word embeddings and document embeddings including BERT, GPT-2 and TF-IDF. The best results were obtained using BERT word embedding as features a CNN as classifier.

In addition to the research problem addressed in this article, the creation and analysis of datasets has been diversified. Most often, a dataset is published with the binary decision about class assignment. The problem of lack of agreement between annotators has often been noted. Typically, there is a lower agreement for a certain subset between the extreme cases (Salminen et al., 2019). These cases are not clear and it depends on the subjective attitude whether they are perceived as hate speech or as still acceptable. Typically, in the annotation process the majority of the annotators for one message is seen as the decision. That means that the dataset does not show which cases are controversial and which are clearly in one class. A recent study has developed a dataset which includes the vote count of up to five annotators. The systems can model the classification problem as a fuzzy assignment (Aroyo et al., 2019). The issue of the generalization of classification results has gained more importance in research. Ultimately, this is crucial for the practical usefulness of these experiments. If the results cannot be transferred to other datasets, the classifiers might also not work well for real data. A thorough analysis of cross dataset performance reveals substantial limitations (Fortuna et al., 2021).

2.1.1. Standard approaches for hate speech detection

For detecting Hate Speech or other variants of problematic online content (aggressive, abusive, offensive, extremist) detection, established text classification algorithms have been applied. In the last years, there has been a shift from lexical representation methods to deep learning models and lately to transformer based architectures (Modha et al., 2021a).

In the last years, a shift in the best performing systems for NLP tasks has been observed. lexical to word embedding, LSTM, BERT, RoBERTa. The performance for different benchmarks does not necessarily translate into higher accuracy measures. The documents contained in the collections and their features are not just too different to lead to similar performance automatically.

The SemEval 2019 Task-5 (Basile et al., 2019) focused on the detection of hate speech against immigrants and women in Spanish and English messages extracted from Twitter. Besides the main binary task to detect hate speech, there was a fine grained task to further classify into aggressive attitude and the target harassed, to distinguish whether a message contains incitement against an individual rather than a group. The best performing system by Indurthi et al. (2019) trained a SVM model with a RBF kernel using Google's Universal Sentence Encoder (Cer et al., 2018) as features. One of the top team (Modha et al., 2018) at the TRAC workshop (Kumar et al., 2018) used a CNN model with FastText word embeddings as input.

Khan et al. (2021) proposed a fine tuned BERT called BERToxic, to locate toxic text spans in a given text. Through experiments, they showed that the two post-processing steps proposed by them improved the performance of their model by 4.16% on the test set.

Apart from pure classification, there are few tools available for end users which apply hate speech classifiers. The existing tools address heterogeneous user populations.

[1] https://hasocfire.github.io/hasoc/2021/index.html

Modha et al. (2020) have realized a system for detecting and visualizing hate speech directly in social media. The system works by visualizing offensive content on Twitter or Facebook comments. It is implemented as a browser based plugin that fetches a comment from the post a user is visiting, send it to an AWS server where it is classified by a CNN classifier hosted on the server into overtly aggressive (OAG), covertly aggressive (CAG), and non-aggressive labels (NAG). Subsequently, it provides a visualization of the results directly within the comments for the user (Modha et al., 2020).

Another system presented an ULMFiT model within a user interface which shows some lexical items to explain the classification (Bunde, 2021). This system is designed for moderators of groups or online media.

Agarwal and Chowdary (2021) proposed an ensemble learning based adaptive model for automatic hate speech detection, improving performance across the cross dataset. The proposed classifier, A-Stacking is an adaptive classifier that uses clustering to conform to the dataset's features and generate hypotheses dynamically. Sharma et al. (2021) proposed a model based on neutralities present in tweets by using deep learning systems.

2.2. Code-mixed text classification

Due to the lack of proficiency in English, many social media users use their local/native language words using Roman script along with English to convey their message. This kind of text is known as code-mixed text. Some of the text analytic work on code-mixed data has been reported in the literature. Joshi et al. (2016) have created the first major English–Hindi Code-mixed dataset for sentiment analysis. The authors have annotated 3879 code-mixed Facebook posts into three polarity classes — positive, negative or neutral. Lal et al. (2019) reported the best results on this dataset using their transformer based model CMSA (Joshi et al., 2016). The reported F1-score is around 0.83. Chakravarthi et al. (2020) created a code-mixed dataset for sentiment analysis in popular Dravidian languages such as Tamil and Malayalam. The top methods on this dataset are based on XLM-Roberta and BERT. Hande et al. (2020) proposed a CodeMixed dataset mainly in Kannada (KanCMD) for sentiment analysis and offensive language identification. The corpus was sampled from YouTube. All these dataset did not consider the context of the social media post while We have prepared code-mixed dataset from the conversation dialogue.

2.3. Conversational and contextual text classification

A standalone tweet can often be hardly interpreted because it is part of a larger discourse and part of a conversation between some users. Using additional context information from the conversation available or from the account is a realistic task for Hate Speech identification. However, only few text classification experiments and datasets considered context for the class assignment. This section shows some which are closely related to our problem. We will also focus on how context is modeled. An early approach used LDA and RNNs (Mikolov & Zweig, 2012). Recursive neural networks were used to capture context within sentences (Park et al., 2018) but less for capturing relations between subsequent messages in social media.

Sentiment analysis is a classification task with some similarities to Hate Speech classification. Ren and colleagues showed that context can be useful for the accuracy of sentiment analysis (Ren et al., 2016). They used a deep CNN which processes text features and context features in parallel before concatenating them in the last fully connected layer. Context was implemented by the words of the tweet that the current tweet is a reaction to, if any. In addition, the words of other tweets of the same author are added as well as the words of tweets on the same hashtag (Ren et al., 2016).

Several approaches use a late fusion of text features and some meta features of the account to facilitate text classification tasks. For example, Wang (2017) has implemented such a model for fake news detection. The last layers concatenate information which was distilled by different systems form the two sources and feeds them into a classifier (Wang, 2017).

The SemEval conference and evaluation initiative introduced the shared task RumourEval in 2019 (Determining Rumour Veracity and Support for Rumours) (Gorrell et al., 2019). RumourEval reacts to the need to consider evolving conversations and news updates for rumors and check their veracity. The organizers provided a dataset of unreliable posts and conversations about those posts. There were 2 tasks, the first one (Subtask-A) was rumor stance prediction containing four classes namely Support, Deny, Query and Comment. The second one (Subtask-B) was about verification of the rumor and it was modeled as a binary classification. The best performing system in subtask B by Li et al. (2019) used word2vec (Mikolov et al., 2013) for word text featuring combined with several other dimensions such as source content analysis, source account credibility, reply account credibility and stance of the source message among others. They concatenated all of these features in one model applied a ensemble approach for classification. The accounts involved in the communication are the source for the context information and model.

Another field of research which uses context is Stance Detection. Stance Detection is concerned mainly with the attitude of someone. The goal is to detect whether a person is in favor of a target or a proposition (Legalization of abortion), is against it or neither of both. SemEval 2016 introduced a shared task for stance detection called Detecting Stance in Tweets (Mohammad et al., 2016). The organizers proposed 2 tasks, Task A which was a supervised task containing tweets from various targets. Task B was a weakly supervised task containing 78,000 tweets associated with Donald Trump. The best performing system for Task A was a baseline model with character N-Gram features and SVM classifier. It delivered the highest Macro F1 score. Out of the submitted systems, the highest Macro F1 was by reached by Zarrella and Marsh (2016). They employed a recurrent neural network initialized with features learned via distant supervision on two large unlabeled datasets. Then they trained embeddings of words and phrases with the word2vec skip-gram method, then used those features to learn sentence representations via a hashtag prediction auxiliary task (Weston et al., 2014). These sentence vectors were then fine-tuned for stance detection on several hundred labeled examples. The authors used five different classifiers, one for each target.

An approach closely related to hate speech detection is the detection toxicity. The notion of toxicity is sometimes used as a more general term than hate speech for problematic content. A dataset was labeled with and without context by crowd workers (Pavlopoulos et al., 2020). The collection was done on Wikipedia talk pages. These are discussion fora for Wikipedia articles to argue about potential improvement that could be made. For the toxicity analysis, 20,000 messages were extracted. Half of them was annotated observing only the text of the message and the other half was annotated with additional context. Context was defined as the parent message and the title of the discussion thread (Pavlopoulos et al., 2020). It needs to pointed out that the parent message might not be the last message preceding the message to the annotated. Interestingly, the percentage of toxic messages is low in this dataset and reaches a maximum of 6 percent. It seems that the messages were collected randomly without using potentially biased words for searching. However, Pavlopoulos and colleagues used two distinct sets of conversations and they compare only the numbers which they have achieved (Pavlopoulos et al., 2020). The performance in both sets is similar, however, this seems no convincing argument that context is not helpful for a classifier. The accuracy of classifiers for different sets of Hate Speech may differ greatly anyway (Fortuna et al., 2021). So the dataset based on Wikipedia talk is a good example for a realistic distribution of offensive content, however, it cannot show whether context is helpful or not.

Table 1
Dataset for problematic content with context and novelty introduced by approach presented here.

Dataset	Context	Problem formulation	Innovation introduced by this paper in comparison
Zarrella and Marsh (2016)	Previous message and account information	detect evolving rumors in social media	Hate Speech detection
Pavlopoulos et al. (2020)	One previous message and title of thread	Toxicity detection in WikiTalk pages	Long contexts over dozens of messages
Menini et al. (2020)	2 to 5 previous messages	Abusiveness detection, context was searched for messages of a plain dataset	dataset originally created for the contextual task

A dataset which conserves the conversational nature of social media messages well and which is related to hate speech is CONAN (Chung et al., 2019). The authors provide a collection of pairs of hate comment and adequate counter-narrative response. This collection was created with the help of skilled experts from NGOs fighting Hate Speech (Chung et al., 2019). However, CONAN was not designed to provide a testbed for conversational classification. The context given is a potential message after the offensive message which would not be available is a realistic setting.

One dataset that was adopted and extended with context information for the notion of abusiveness (Menini et al., 2021). A drawback of this approach is the creation of the dataset and the modeling of the problem. The data was collected based on an existing dataset without contextual information. For all tweets, the text was used to search them and if they were found, the authors tried to extract the previous messages. For all tweets, for which this was successful, the preceding messages were downloaded as context. This leads to a situation in which the context size is very different from instance to instance. The authors report that around 45% of the hateful tweets had one preceding tweet as context and another 45% had between 2 and 5 preceding tweets. Only some 10% had more than 5 tweets available as context. Two rounds of annotation with the same annotators after a longer time period were conducted. In the first round, annotators saw only the tweet and in the second round they were also given the context. Applying this methodology, almost half of the tweets which were annotated as abusive were labeled as non-abusive once context was available (Menini et al., 2021). This overview on the state of the art shows that the generation of a new conversational dataset which is initially designed with a fixed model of context is necessary for advancing the area. In particular, the collection of the conversational context is not clearly described. Table 1 present the comparison of our approach to the 3 most similar works.

2.4. ICHCL task @HASOC'21

We have offered Identification of Conversational Hate-Speech in Code-Mixed Languages (ICHCL) task[2] at HASOC forum in FIRE'21 conference at this occasion, the newly designed ICHCL dataset was presented to the NLP research community. Overall, 15 research teams participated in the shared task. The reported macro F_1 score ranges around 0.49 to 0.73 (Modha et al., 2021b). The ICHCL task's top team, team MIDAS (Farooqi et al., 2021) developed ensembles of three transformer models, namely IndicBERT, Multilingual-BERT and XLM-RoBERTa and reported macro-F1 score around 0.729. The authors concatenate posts to represent the conversational dialogue. The next two teams, Super Mario (Banerjee et al., 2021a) and IIIT Hyderabad (Kadam et al., 2021a) used models based on XLM-RoBERTa and reported a macro F1 score around 0.71 and 0.70 respectively. The majority of the teams used different variants of BERT such as multilingual BERT, Indic BERT for the classification. Team PC1 (Modha et al., 2021b) adopted a completely different approach. The authors converted text in Devangari script to ASCII characters. The author claims that this will work for any language. TF-IDF was used to represent the character n-grams on the normalized text and classification was done using Logistic Regression. These results represent the state of the art performance for contextual Hate Speech identification.

Table 2
ICHCL Dataset : Dataset statistics of Train and Test set.

Dataset	# Twitter posts		#Comments the posts		#Replies	
	HOF	NONE	HOF	NONE	HOF	NONE
Train	49	33	1820	1958	972	908
Test	9	7	433	416	253	230
Total	58	40	2253	2374	1225	1138

3. ICHCL dataset

This section will elaborate on the data collection. It will present the collection, annotation, the definition of a story or tweet, as well as the data structure. To our best knowledge, this represents the first data collection which was designed to contain conversational context from the beginning. This dataset is named ICHCL (Identification of Conversational Hate-Speech in Code-Mixed Languages) dataset and was presented to the research community at HASOC Forum[3] at the FIRE'21 Conference.[4] For corpus creation, we focused on some common controversial societal issues related to gender discrimination, religious intolerance and the COVID19 crisis. The sampling and annotation of social media conversation threads are very challenging. We have chosen controversial stories on diverse topics to minimize the effect of bias. We have hand-picked controversial stories from the following topics that have a high probability of containing hate, offensive, and profane posts. The controversial stories are as follow:

1. Twitter Conflicts with the Indian Government on new IT rules.
2. Casteism controversy in India
3. Charlie Hebdo posts on Hinduism
4. The Covid-19 crisis in India 2021
5. Indian Politics
6. The Israel-Palestine conflict in 2021
7. Religious controversies in India
8. The Wuhan virus controversy

Table 2 illustrates the statistics of the ICHCL dataset presented at the HASOC forum at the FIRE conference. Overall, class labels are distributed evenly and the dataset looks balanced. We have splitted the ICHCL dataset into the training and testing dataset in the ratio of 80:20. 80% of the overall data was used as a training dataset. Similarly,20% of the overall dataset was used for testing.

[2] https://hasocfire.github.io/hasoc/2021/call_for_participation.html

[3] https://hasocfire.github.io/hasoc/2021/ichcl/index.html

[4] http://fire.irsi.res.in/fire/2021/home

Table 2 shows the statistics of the ICHCL dataset. The dataset can be downloaded from this link[5]

In the following Section 3.1, we will discuss the methods used to sample the conversational data from Twitter. Annotation guidelines, inter-annotator agreement details, and the data structure used for the data distribution are included in Section 3.2

3.1. Sampling of conversation dialogue

We have manually downloaded potential offensive conversational dialogues from Twitter, using a scraper developed with the Twitter API and the Selenium browser automation tool. This tool helped us to scrap Twitter posts, comments on Twitter posts and replies to each comment.

The tweets collected are written in English and Hindi and used mixed scripts. This is typical for multilingual societies like in India. Users of mobile phones do not always switch to an adequate script for each language. For example, they might use the Devanagari script (for Hindi) when writing English words within a message or even for a full English message and vice-versa.

3.2. Data annotation

Annotating the conversation dialogue is a challenging task. Therefore, we have built our own annotation platform to label the posts. The platform can be seen in a sample video.[6] Each tweet which includes posts, comments, and replies is annotated either HOF or NONE.

- (HOF) Hate and Offensive — This tweet, comment, or reply contains Hate, offensive, and profane content in itself or supports hate expressed in the parent tweet
- (NOT) Non Hate-Offensive — This tweet, comment, or reply does not contain any Hate speech, profane, offensive content

To guarantee a high level of quality, no crowd workers were employed for the task, but only the authors themselves and a pre-final year student annotated the whole dataset. The allocation for annotations was done in three phases. In the first phase, we assigned conversations to all annotators randomly. After the first phase of annotating, the annotated comments and replies were again assigned randomly to a different annotator. After checking the agreement between the two annotators, the conflicts were resolved during a third phase. Only those tweets (comments and replies), for which the label of the first and second annotator differed, were assigned to a third annotator randomly. The third annotation decides the final label of the tweet.

The interrater agreement is shown in the Table 3. It can be considered as substantial and lies in the range of other hate speech datasets.

Table 3 contains the details of the inter-annotator agreement. The third column (agreement) contains the individual agreement between each individual annotator. The fourth column (Cohen's co-eff) is the cohen kappa score, which is more reliable since it considers the expectation of an agreement based on hypothetical likelihood of chances of agreement rather than just the percentage of agreement.

We used JSON data structure to distribute the data to the community. Fig. 3 shows a snapshot of one of the conversational dialogues that contain a parent tweet with a single comment and a single reply. This structure has been followed throughout the entire dataset.

We have distributed the dataset in JSON format. Fig. 3 shows a sample conversation.

[5] https://hasocfire.github.io/hasoc/2021/dataset.html
[6] https://www.youtube.com/watch?v=DJq7OGdWRDE

Table 3
Inter-annotator agreement between annotators.

Annotator1	Annotator2	Agreement	Cohen's co-eff
A1	A5	0.6957	0.3750
A1	A2	0.8415	0.6822
A3	A4	0.7156	0.4348
A4	A5	0.8988	0.7982
A2	A4	0.6877	0.2920
A2	A3	0.7099	0.3730
A6	A1	0.6047	0.3162
A3	A5	0.8889	0.7805
A6	A2	0.5785	0.2353
A2	A5	0.8800	0.3590
A1	A4	0.6492	0.3274
A6	A5	0.5746	0.2592
A1	A3	0.6679	0.3261
A6	A4	0.6109	0.1591

4. Experimentation setup

For designing the classification experiments, we focused on evaluation using diverse machine learning and deep learning methods combining them with various text representation techniques to produce an extensive set of results. The techniques for obtaining the features are discussed in Sections 4.1.1 and 4.1.2 and the classification models are discussed in Section 4.2.

The crucial part we had to focus on in solving this problem was how to incorporate the context of a parent into comments and replies. We experimented with several different techniques for this. They are discussed in Section 4.1.3.

The use of highly diverse approaches provides a solid base for a deeper analysis of comparison between combinations of models and featuring techniques to deduce which combinations can adapt to the problem.

4.1. Features for text representation

Extracting features and computing representation for this contextual classification problem requires a more open approach than many traditional NLP classification problems. Additionally, the data represents real-world multilingual and code-mixed text to further complicate the task. We have used a mix of basic and advanced embedding techniques to tackle the problem.

Section 4.1.1 discusses traditional bag of words based methods and Section 4.1.2 discusses advanced featuring methods based on deep learning which were used for experimentation.

4.1.1. Bag-of-Word(BoW) weighted by TF-IDF

Term frequency and inverse Document Frequency (TF-IDF) are a popular weighting technique used in lexical systems. Words are used to represent the content of documents (or tweets) and words with lower frequency are given a higher weight. Typically, term frequency is multiplied with the inverse document frequency but there are also many variants. The TF-IDF (Aizawa, 2003) document weighting technique was prevalent in the early 2010s. Since then, several newer embedding techniques have emerged but we still experimented with TF-IDF, as the results for the problem we are addressing in this research may benefit using TF-IDF at the lexical level. Despite issues when the lengths of the documents are very different, TF-IDF is still often used as a baseline. The weights were generated by using the popular library SciKit-Learn (Pedregosa et al., 2011). A minimum document frequency of 5 was taken for vectorization to decrease the length of representation vector to avoid underfitting and faster computations. The length of the vocabulary taken into consideration was 2511. This is also the length of the representation vector obtained.

```
{
    "tweet": "A picture of the Israeli Terrorist while
    ↪ trying to arrest a Jerusalemite young man near the
    ↪ Al-Amoud Gate. #FreePalestine #JerusalemFightsBack
    ↪ #WakeUpMuslimUmmah https://t.co/nSTrkI1Mtv",
    "tweet_id": "1391754900896468996",
    "comments": [
        {
            "tweet_id": "1391757510198009858",
            "tweet": "@user1 @user2 follow back",
            "replies": [{
                "tweet_id": "1391864183298220035",
                "tweet": "@user2 @user1 Ghairat kitnai ki
                ↪ baichi?"
            }]
        }]}
```


Fig. 3. Data Structure of Conversation.

4.1.2. Embeddings

Other than traditional bag of word based methods, recently shallow encoding method such as word2vec and Fasttext have been introduced. Furthermore, attention based models have been introduced recently which led to substantial improvement. We have leveraged these two encoding techniques and described the experimental setup in the following subsections.

4.1.2.1. FastText. As a popular version of modern word embeddings, we used FastText (Bojanowski et al., 2016) which is especially suited for text from social media. Word embeddings generate an embedding of a dimensionality between 50 and 300 for each word. These embeddings are trained with large text corpora such that words appearing in similar contexts obtain similar embeddings. FastText can also capture information at the sub-word level. For example, suppose "Class" is an unknown word in the FastText dictionary. To represent this word, FastText can tokenize the word class into multiple subwords like "Cla", "lass", "ss", "C", etc. This ability of FastText can be utilized to the sub-word and even character level for the code-mixed tweets. This can be highly beneficial for recognizing patterns. The length of the FastText word embeddings which we chose for experiments was 300.

4.1.2.2. BERT. BERT (Devlin et al., 2019) is a recent language model based on a transformer architecture and a masked training scheme. It typically obtains better performance than most of the other existing embedding techniques for many NLP tasks. It also includes sub-word levels information just like FastText. It is trained to analyze dependencies in text in both directions and then calculates the embedding for a given word or sentence.

4.1.2.3. Fine tuned SentBERT. We used transfer learning to train a sentence similarity transformer using Supervised Contrastive Loss (Khosla et al., 2020). For traditional cross entropy loss, the main objective is to calculate the encoded representation so that they can be linearly separable. However, supervised Contrastive Loss can be called "greedy". The main aim in supervised Contrastive Loss is to train the encoder as a Siamese network such a way that the encodings of samples with same label come closer to each other in the geometric space and distance increases in the samples with different labels. This loss function has led to the best results for Imagenet after it was introduced.

$$L(i,j) = \frac{-1}{2N_{\tilde{y}_i} - 1} \sum_{j=1}^{2N} \mathbb{1}_{j \neq i} \mathbb{1}_{\tilde{y}_i = \tilde{y}_j} \log \frac{\exp \frac{sim(Z_i, Z_j)}{\gamma}}{\sum_{k=1}^{2N} \mathbb{1}_{K \neq i} \exp \frac{sim(Z_i, Z_j)}{\gamma}} \tag{1}$$

For our experiments, we have used a Sentence-BERT model (Reimers & Gurevych, 2019).[7] A Sentence-BERT is a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. The pretrained transformer which is publically available, is generally used for IR tasks however we have leveraged it to work on our task. We have trained this transformer model as a siamese network. To make pairs, we suppose that if the child and its immediate parent have similar labels than they are similar else they are dissimilar. We trained this model for 10 epochs. Using this transformer model, a document embedding of length 768 was retrieved for each text level.

4.1.3. Context representation

In traditional problems it is assumed that all the samples are independently and identically sampled from some probability distribution. So each sample we are supposed to classify is independent of the rest of the samples in the dataset. However in this problem, the comments and replies are not independent of their parent text. Still, we conducted experiments by considering them independent and presented the results in Section 5.1 which we will further discuss in that subsection.

In the following subsections, we introduced multiple ways of context representation for text and the feature vectors calculated by any of the above mentioned text representations methods. Since, to represent context, multiple levels of text is being fused we used Context Representation and Fusion interchangeably in the rest of the paper.

[7] https://www.sbert.net/

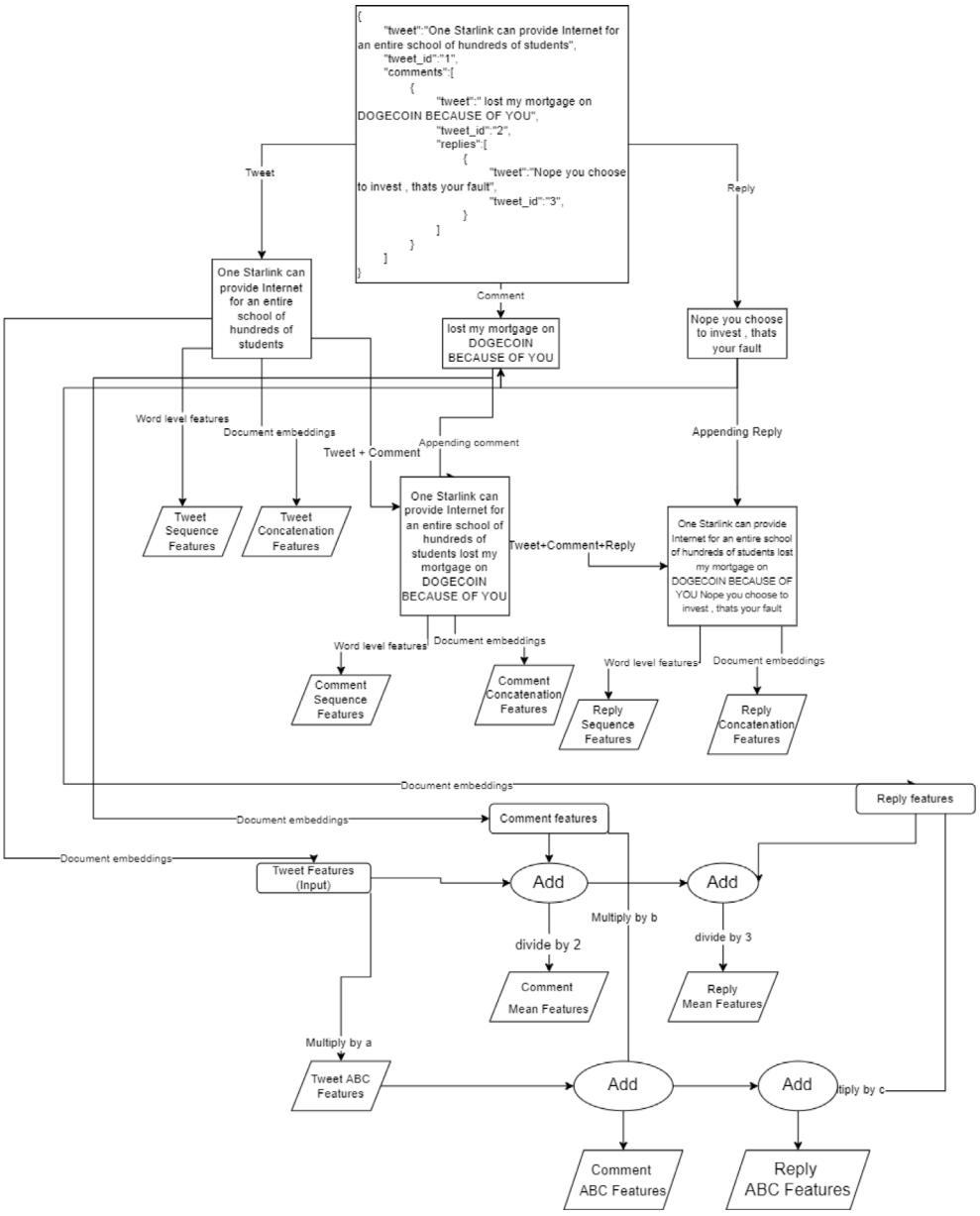

Fig. 4. Flowchart of context representation techniques.

Fig. 4 is a visual representation of how all the context representation techniques work. In the flowchart, rectangle entities denote a text level, ovals denotes basic operations, squircle entities denote intermediate document independent embeddings of each text level that is to be processed and parallelogram denotes the final features that will be used for classification.

4.1.3.1. Concatenation. Concatenation is one of the most widely used fusion techniques used. It was mentioned several times in the state of the art section (e.g. Wang, 2017). We also applied concatenation. However, we are dealing with three different levels of data. So, if we concatenate all embeddings one after another the length of vectors will be different. So instead, the text is concatenated and then vectorized. That way, the length of the embedding vector remains uniform for the initial tweet and the conversational context.

For our experiments, all the comments of a tweet were concatenated at the end of the tweet. So the comments are represented by "< Tweet > <Comment >". And in case the comments also have replies, these replies were concatenated at the end of tweet and comment. Consequently, the replies leads to the following representation: "<Tweet > < Comment > < Reply >". After concatenation, the text is vectorized. The length of vector for TF-IDF is 2511. For BERT, the document embeddings of length 768 and for FastText, the mean of all the word embeddings was calculated using the same weight for all three elements. The length of each vector is 300.

4.1.3.2. Mean. For this part, the 3 text levels were vectorized separately. For the tweet, the same embedding were assigned as features. For comment, mean of comment's embedding and tweet's embedding was taken. For reply, mean of reply's embedding, comment's embedding and tweet's embedding was taken.

The length of the document embedding for TF-IDF is 2511. For BERT, it is 768. The mean of document embeddings of separate text levels was calculated to obtain the features of a reply or comment. For FastText, there is no notion of a document embeddings. It can only provide word embeddings. Hence to get document embeddings for a text level the mean of all word embeddings was calculated and than the mean of different level of embedding was calculated to obtain features for the level that is needed to be classified. The length of each vector for FastText is 300.

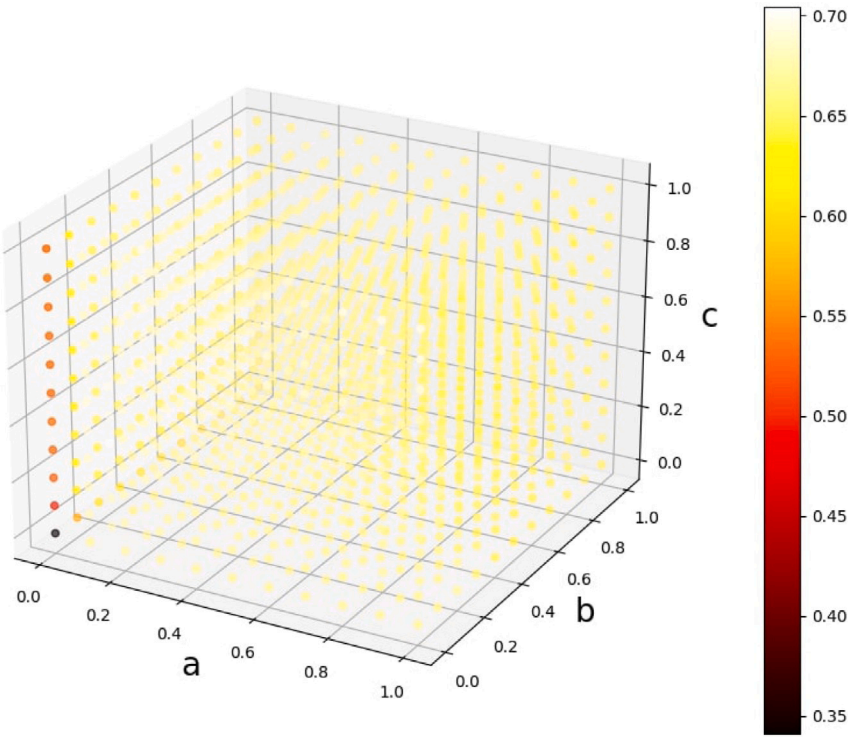

Fig. 5. Grid search on weights.

4.1.3.3. Sequence. For this fusion strategy, the text levels are concatenated just like in the concatenation fusion. But instead of document embeddings, the tweets were tokenized and then the word embeddings of the token were concatenated to form a matrix for each sample. The resulting sequence of vectors is fed into a neural network as an input. Post padding was applied to ensure equal size. The size of each matrix for BERT is (768,168) and for FastText it is (300,117).

4.1.3.4. ABC weighting. A weighted embedding technique can give different importance to the three elements of tweet and context. Each level of the tweet is assigned a weight which can be used as a hyperparameter. A is the weight for the tweet, B is the weight for the comment and C is the weight of reply to the comment. The weights are applied to the level that is being classified and all the corresponding parent levels.

$$\begin{aligned} embedding &= a * e(t) + b * e(c) + c * e(r) \\ where,\ e(t) &= document\ embedding\ of\ tweet \\ e(c) &= document\ embedding\ of\ comment \\ e(r) &= document\ embedding\ of\ reply \end{aligned} \tag{2}$$

Each of the weights ranges from [0,1] with a step length of 0.1. For example, if the triplet (0.2,0.3,0.9) represents the weights at a given iteration, then in the next iteration the following triplet of weights is used: (0.2,0.4,0.0). A total of 1331 value triplets were used for the experiments with the weights. The length of the representation vectors were similar to the ones of the Mean representation. Fig. 5 shows the final grid search results for all the weight triplets. It is discussed in detail in Section 5.3.

4.2. Classification models

For the experiments, we applied several heterogeneous classification methods in order to achieve a broad coverage for benchmark results.The Section 4.2.1 discusses setup for different traditional ML models and 4.2.2 discusses the deep neural network models.

Table 4
List of traditional classifiers.

Classifier	Abbreviations	Hyperparameters
Logistic regression	LR	Default
Naive Bayes	NB	Default
K-Nearest Neighbors	KNN	Neighbors = 3
Support Vector Machine (Linear/Radial)	SVML and SVMR	Regularization = 0.5, Kernel = Linear,Radial

4.2.1. Tradional classifier models

Table 4 presents the set of traditional classifier models we experimented on, with their hyperparameters and abbreviations for the classifiers that we will be using for the rest of the paper. For the experiments reported here, we used the implementations from the SciKit-Learn library.

4.2.2. Neural models

4.2.2.1. Feed forward neural network models. A fully connected neural network trained with a backpropagation algorithm was implemented using Tensorflow (Abadi et al., 2015) and Keras. The model has 10 layers. Layers 1 through 4 are dense layers with 1536, 2048, 2560, 2560 neurons respectively and they use the ReLu activation function. A dropout layer with 0.5 probability of dropout follows. The four more dense layers with 2560, 2560, 2048 and 1536 neurons with the ReLu activation function follow. The last layer is a dense layer leading to one neuron and a sigmoid activation function for the actual classification.

The network was trained using binary_crossentropy as loss function, the Adam optimizer with learning rate of 0.0003 for 20 epochs. This neural network was trained with BERT-Mean, BERT-Concatenate, TFIDF-Mean and TFIDF-Concatenation as features.

4.2.2.2. Long–short term memory networks(LSTM). Long–short-term memory models are using a metaphor of the human cognitive system. They include a long-term model of the data which is obtained for a longer time and which tries to model longer distance dependencies. In

addition, a short term element enables the parallel processing of short distance dependencies.

This LSTM model has 9 layers. Layers 1 and 2 are LSTM layers with 64 units followed by a dropout layer with 0.5 probability. Layer 4 and 5 are LSTM layers with 64 and 128 units respectively again followed by a dropout layer with 0.5 probability. Layer 7 and 8 are LSTM layers with 128 units again followed by a dense layer with one neuron and Sigmoid activation for classification.

The network was trained using binary_crossentropy as loss function, the Adam optimizer with learning rate of 0.005 for 50 epochs. This neural network was trained with FastText and BERT word embedding sequences. Also this was used a base case to compare the whether the inclusion of context helps in classifying or not.

4.2.2.3. Convolutional neural network(CNN). Convolutional neural networks (CNNs) were originally developed for image processing and can link local regions to features. These local features are extended to larger patterns over each layer. The convolutions are small matrices which can extract patterns and which adapt through the learning process.

The CNN we used a setting similar to the work produced by Yoon Kim (Kim, 2014b). In this neural network setting, the kernel size defines the number of consecutive word vectors we want to analyze. A CNN block has 3 layers. A 1D convolution layer with 128 filters, valid padding and ReLu activation. The kernel size is set to the number of consecutive word vectors we want to analyze. The convolution layer is followed by a Global Max Pooling and a batch normalization layer.

For our training we used three such blocks with the kernel sizes 2, 3 and 4, respectively. The features that are attained at the end of the third block, are concatenated into one tensor and then it is followed by three dense layers with 256 and 128 neurons with ReLu as activation function and finally with one output neuron and Sigmoid as activation function.

This neural network was trained with FastText and BERT word embedding sequences. Similar to the LSTM model, is was also used to analyze the inclusion of context.

5. Results and analysis

This section will discuss the exhaustive results of combining text and context representation techniques and a set of classifiers. Section 5.1 presents the importance of conversational context on classification results, and exhaustive results of various context processing pipelines are presented and discussed in Section 5.2. Section 5.3 delves a little deeper into error analysis by analyzing various misclassification patterns that tend to happen while classifying using all the pipelines. Furthermore, Section 5.4 discusses the outcomes of the research objective set in the Introduction section. The code for best performing systems is available on GitHub.[8]

5.1. Importance of conversational context

A Twitter conversation or social media dialogue consists of a parent tweet (posts), one or more comments to the parent tweet, and one or more replies to the comment. There are two ways to represent the comment tweet and tweets tweeted to reply to the comment. The first way is to represent a comment tweet or reply tweet independently. The second way is to represent it with the context of its parent tweet. The same is discussed in Section 4.1.3:Context Representation.

Before we present the result on the state of art classifiers, we want to prove our initial hypothesis of whether considering the tweet's context improves the classification model's performance. To prove this, we have trained two CNN and two LSTM neural network classifiers to test the classification using context and without context. So for 1 LSTM and 1 CNN, the comments and replies will be trained with texts independently, which is the first method mentioned in the paragraph above. Moreover, 1 LSTM and 1 CNN network were trained while considering the context. For context, we used the sequence featuring scheme, and for the independent levels, the tweets were tokenized without concatenation, and the embedding matrix of the text alone was used to obtain the features. After training and evaluating, a comparison was carried out. Table 4 shows the results. It can be observed that the context of the parent tweet helps the classifier to make better predictions compared to not considering the context.

Table 5
Results on ICHCL dataset: With and without context.

Model	Context considered	Macro F1 score on test set	10-Fold macro F1
CNN	No	0.621	0.619
CNN	**Yes**	**0.623**	**0.708**
LSTM	No	0.544	0.511
LSTM	**Yes**	**0.597**	**0.691**

From the table, it seems that LSTM performance increases significantly after considering the context. Since a LSTM can capture longer dependencies better more effectively on than CNN and would not be efficiently utilized with shorter texts. Without considering context, most of the lower level text (comments or replies) were either a few emojis or some affirmative or supporting words (for example *IE yes*, *LoL*, *exactly*, etc.). Further analysis revealed that the mean number of tokens generated by BERT without context is 39.39. Considering context is rose to 74.93. The LSTM gives significantly better results when context is used.

5.2. Experimental results on context processing pipelines

Table 6 is a full list of the exhaustive results of all combinations of classifiers and features for which we have carried out experiments. 10-Fold Macro F1 is the mean of macro F1s evaluated over the 10 training folds. The results are sorted in descending order of 10-fold Macro F1.

For ABC weighting, to choose the best weight triplet we used a grid search method for all the 1331 triplets using TFIDF and training a Logistic Regression algorithm. Fig. 5 displays the results achieved using different triplets for this method. The figure describes the F1 macro score achieved at all the weights. It shows a scatter plot of the grid search across all the triplets of the weights we experimented with using the ABC weighting context representation scheme.

After experimentation, we observed that triplets [0.1,0.1,0.3], [0.1,0.1,0.5], [0.1,0.1,0.4] and [0.2,0.1,1] were the top 4 triplets with no more than 0.01 difference in F1 Scores. This shows that the original tweet should be included in the processing but that it should be given a lower weight than the response which needs to be classified.

As mentioned in Section 3, we organized a shared task (Modha et al., 2021c)[9] and made the dataset public. 16 teams participated in this shared task and we received a number of submission. Table 7 contains the comparison table of the top 5 team submissions. As Table 6 show, our system using Sentence-BERT and the Sequence context representation with LSTM as classifier obtains better performance than the best submission from the shared task.

The results show that the effect of the feature construction and assigning weights to the different text parts is strong. The sentence transformer SentBERT which was fine tuned on the dataset using Contrastive Loss and ABC fusion have led to the best performance of 0.89 F1 measure. The machine learning classifier has not such a strong impact. Four classifiers using SentBERT and ABC fusion perform between 0.808 and 0.79. This means, that assigning a lower weight to

[8] https://github.com/HirenMadhu/SentTrans-KNN-ICHCL

[9] https://hasocfire.github.io/hasoc/2021/ichcl/index.html

Table 6
Exhaustive experiment results with set of classifiers and text representation schemes on ICHCL dataset.

Classifier	Representation	Fusion	10-Fold macro F1	Macro F1 on test dataset
LSTM	SentBERT	Sequence	**0.928**	**0.892**
CNN	SentBERT	Sequence	0.920	0.879
KNN	SentBERT	ABC	0.874	0.808
SVML	SentBERT	ABC	0.836	0.803
SVMR	SentBERT	Mean	0.83	0.706
SVML	SentBERT	Mean	0.828	0.69
LR	SentBERT	ABC	0.824	0.791
KNN	SentBERT	Mean	0.82	0.652
LR	SentBERT	Concatenation	0.816	0.671
LR	SentBERT	Mean	0.814	0.683
SVMR	SentBERT	ABC	0.813	0.801
NB	SentBERT	ABC	0.804	0.792
KNN	SentBERT	Mean	0.802	0.605
NB	SentBERT	Concatenation	0.751	0.68
NB	SentBERT	Mean	0.739	0.683
LR	TF-IDF	Concatenation	0.716	0.623
LR	TF-IDF	Mean	0.712	0.623
ANN	TF-IDF	Mean	0.707	0.639
SVMR	TF-IDF	Concatenation	0.705	0.626
SVMR	TF-IDF	Mean	0.704	0.605
ANN	TF-IDF	ABC	0.702	0.613
SVML	TF-IDF	Concatenation	0.702	0.614
SVMR	TF-IDF	ABC	0.70	0.59
ANN	TF-IDF	Concatenation	0.698	0.605
SVML	TF-IDF	Mean	0.697	0.595
SVMR	FastText	Concatenation	0.696	0.612
SVMR	FastText	Mean	0.693	0.633
LR	BERT	Mean	0.693	0.654
SVML	BERT	Mean	0.69	0.66
LR	FastText	Concatenation	0.69	0.636
LSTM	FastText	Sequence	0.691	0.534
CNN	FastText	Sequence	0.686	0.552
LR	TF-IDF	ABC	0.684	0.625
SVML	FastText	Concatenation	0.684	0.611
LR	FastText	Mean	0.682	0.614
SVML	FastText	Mean	0.68	0.611
LR	BERT	ABC	0.671	0.660
SVML	BERT	ABC	0.671	0.668
SVML	BERT	Concatenation	0.665	0.63
NB	TF-IDF	Mean	0.663	0.557
ANN	BERT	Mean	0.66	0.558
NB	TF-IDF	Concatenation	0.66	0.594
LR	BERT	Concatenation	0.657	0.629
ANN	BERT	Concatenation	0.652	0.573
SVMR	SentBERT	Mean	0.651	0.664
SVMR	BERT	Concatenation	0.637	0.649
SVMR	BERT	ABC	0.636	0.657
LR	TF-IDF	ABC	0.625	0.645
SVML	TF-IDF	ABC	0.623	0.579
NB	BERT	MEan	0.585	0.652
NB	FastText	Concatenation	0.576	0.509
NB	FastText	Mean	0.57	0.542
NB	BERT	Concatenation	0.489	0.55

Table 7
Peer comparison with top teams in HASOC ICHCL 2021.

Rank	Team name	Macro F1
–	**Our System(SentBERT+LSTM+Seq)**	**0.892**
1	MIDAS-IIITD (Zaki et al., 2021)	0.7253
2	Super Mario (Banerjee et al., 2021b)	0.7107
3	PreCog IIIT Hyderabad (Kadam et al., 2021b)	0.7038
4	rider (Mundra et al., 2021)	0.6890
5	Hasnuhana	0.6866

the main tweet and comment gives a robust result for our classification problem. Neither a basic BERT representation nor a FastText word embedding representation can reach a performance of 0.7. Even a TFIDF representation model gives stable performance of 0.62. As mentioned, the best results with ML classifiers were observed for Sentence-BERT paired with ABC fusion and KNN as classifier. This is due to the Supervised Contrastive Loss used to fine tune the transformer model. As the transformer is trained on this loss such that hate speech text comes closer to other hate speech text when encoded, which intuitively assists the K-Nearest Neighbors algorithm to classify more efficiently compared to other algorithms. Upon further analysis, it was found that the mean distance between 2 samples with same label tends to be smaller than the mean distance for 2 samples with different labels. For example, in this dataset with the ideal ABC weights, the mean euclidean distance in the geometric space between 2 positive samples was 3.87, for 2 negative samples it was 3.78 but for samples with different labels it was 4.67. This shows that KNN, even though being a primitive algorithm can perform very well with featuring and fusing methods that resonates with its strong points.

However, the best results were achieved using an LSTM model with the SentBERT features and sequence as context representation techniques. There is an increase of 8% in the F1 macro of the compared to KNN with ABC weighting technique. This goes to show that with an

optimized feature extractor, we can boost the performance of classifiers by a lot margin.

In the top ten runs, we can see that the following technologies were present most often:

- Representation: Sentence BERT was present 10 times
- Context Representation: ABC was present 3 time, Sequence was present 2 times, concatenation once and Mean was 4 times
- Classification Algorithm: CNN and LSTM are present once, KNN was present twice, SVM 3 time and LR 3 times

It needs to be noted that the SentBERT method delivered very robust results.

The experiments show that the impact of the parts considered should be carefully chosen. It can be seen that weighting has a great potential. Even with a feature technique like TFIDF which is not state of the art and not based on deep models, a performance above 0.6 can be achieved using the ABC method after parameter optimization.

5.3. Error analysis

In order to better understand the model and potential factors influencing the performance, we conducted an error analysis. For that goal, we considered tweets which were assigned the wrong label in the two experiments with the best performance. We considered the presence of named entities, the sentiment score and number of times a text was misclassified for all the combinations.

For the analysis of the sentiment, we computed sentiment scores of the tweets using a huggingface model based on RoBERTa.[10] Barbieri et al. (2021) which computes positive, negative and neutral sentiment scores. Using these sentiment scores we can make some high-level assumptions regarding what kind of patterns of sentiment and labels are most often misclassified. Identifying such patterns can help us understand what underlying characteristics of context are strenuous to classify accurately.

It can be seen that the tweets with negative sentiments were most often misclassified. They were followed by tweets with neutral sentiment and finally positive sentiment.

For contextual analysis, comments with hate speech can have a non-hate parent or comments with non-hate can have a parent with hate speech. When the sentiment of these patterns is analyzed, it can be seen that the comments having positive sentiment were misclassified the least often as compared to tweets with neutral or negative sentiment. A similar pattern was observed when a reply with hate speech had non-hate parents or a reply with non-hate parents contained hate speech.

For named entities, there were no significant patterns except for replies. When no named entities were present in the text hate speech was less accurately classified compared to when named entities were present. For non-hate replies, when named entities were present text was less accurately classified compared to when no named entity was present.

5.4. Outcome of research objectives

- *To develop a hate and offensive speech dataset from the social media conversational dialogue.*
  We have sampled the hate and offensive speech dataset from Twitter and named it ICHCL (Identification of Conversational Hate-Speech in Code-Mixed language) dataset. We offered ICHCL a shared task at the HASOC forum[11] at the FIRE'21 conference.[12] The newly designed ICHCL dataset was presented to the NLP research community. Overall, 15 research teams across the world participated in the shared task. The reported macro F1 score ranges from 0.49 to 0.73 (Modha et al., 2021b). We believe that we will get a reasonable response from the community for the new task and dataset.
- *To capture the context of social media communication to improve the performance of the offensive text classification.*
  The second objective of this paper is to ensure that capturing the context can improve the performance of the hate and offensive text classification system. In response to this objective, we trained CNN and LSTM -based classifier models to test the classification using context and without context. For context inclusion, we used the sequence featuring scheme. For the condition of no context, the tweets were tokenized without concatenation, and the embedding matrix of the text alone was used to obtain the features. After training and evaluating, a comparison was carried out. Table 5 shows the results. It can be observed that the context of the parent tweet helps the classifier to make better predictions compared to not considering the context.
- *RO-3. To find the optimal way to represent conversational dialogue in a fixed-length vector.*
  We proposed and experimented with various ways to represent the context of a conversational dialogue discussed in Section 4.1. We proposed context representation techniques such as concatenation, mean, sequence, and ABC weighting scheme. Moreover, we have used a wide variety of features. The results of the extensive combinations of these techniques have been presented in Table 6:Exhaustive Experiment Results with a set of classifiers and text representation schemes on the ICHCL Dataset. It is noticeable that the Sentence BERT model, which was fine-tuned on the ICHCL dataset, works very well with all kinds of context representation techniques.
- *RO-4 To experiment with various classifiers to obtain the best performance on the proposed ICHCL dataset*
  As mentioned, we have used an extensive set of classifiers and context representation techniques which led to the extensive set of results presented in Table 6. It is noticed that the fine-tuned Sentence BERT works best on the ICHCL dataset. All the top 10 results in Table 6 have Sentence BERT as a text representation technique. Even with traditional ML classifiers, the ABC weighting technique has yielded high results. We can infer that with optimal features, even simple traditional classifiers such as KNN or SVM perform substantially better than a base BERT model paired with neural networks such as CNN or LSTM. However, when LSTM or CNN are trained with text representations from an optimized sentence BERT model, it leads to better results than base BERT text results.

6. Conclusion and outlook

This study has shown that including conversational context can improve text classification results when processing social media data. Taking the discourse and the entire conversation into account can improve Hate Speech recognition. For this task, basic BERT representations do not deliver the best accuracy. Our most promising results were obtained with the Sentence-BERT and weighting method. This shows that optimized transformer models can still improve the classification results and assigning weights to different levels also has a positive impact on the performance.

In future work, it will be necessary to develop more datasets with context and conversation information and further refine the notion of the necessary context. Further research needs to investigate how many and which previous messages are useful to be considered as context for the analysis of a social media post. The ABC weighting suggested in this paper could be further refined into models which learn to observe the course of a conversation and identify the text of

[10] 'cardiffnlp/twitter-xlm-roberta-base-sentiment'.
[11] https://hasocfire.github.io/hasoc/2021/index.html
[12] http://fire.irsi.res.in/fire/2021/home

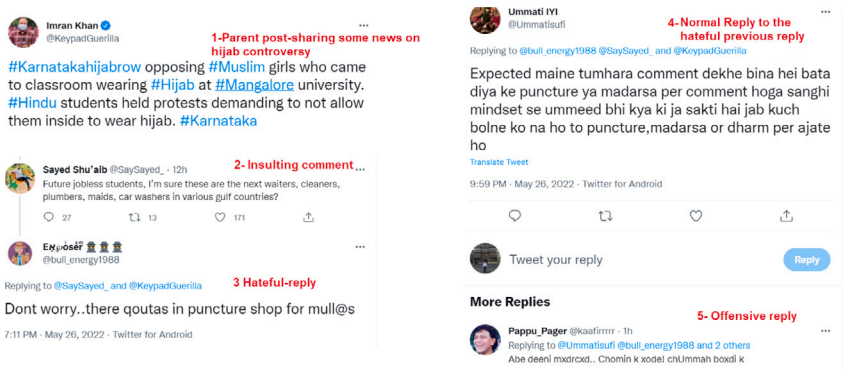

Fig. 6. Example for potentially hateful development of conversation.

previous messages which are of particular importance. This will also require the analysis of references within text. One crucial task could be to find the moment in a conversation when hatefulness occurs and characterize such events. For this goal, We will propose a statistical or neural model that can estimate the probability of the dialogue that has the potential to become hateful/offensive in the upcoming time-series step. The probability of dialogue which is initiated by a political leader has a higher chance of becoming hateful than that of a scientist or a sportsperson. Fig. 6 shows a conversation that starts with some controversial topic like Hijab and soon becomes abusive in the next comment tweet and replies become hateful and offensive in the next 3 tweets.

Due to the huge volume of tweets, it might be impossible to filter each and every tweet. We believe that future models will assist social media in becoming more vigilant about such conversations. Further ideas for such context analysis of linear text can come from research on other domains like sequential analysis and temporal patterns.

Furthermore, it is necessary to create datasets for other languages and other social media platforms. Also, more advanced classification systems using context information are likely to be developed. Hate Speech has also a strong multimodal component. Only when considering image and text, the aggressive nature of a post can be detected sometimes and first datasets have been developed in this direction (Menini et al., 2020). In future work, multimodal context within a conversation can be considered as well.

CRediT authorship contribution statement

Hiren Madhu: Methodology, Software, Writing – original draft. **Shrey Satapara:** Visualization, Software. **Sandip Modha:** Conceptualization, Writing – original draft. **Thomas Mandl:** Conceptualization, Supervision, Writing – review & editing, Validation. **Prasenjit Majumder:** Supervision, Writing – review & editing, Validation.

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

We are thankful to an anonymous reviewer of the Expert Systems With Applications journal who inspired us to formulate this problem during the reviewing process of a previous paper Modha et al. (2020).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/ software available from tensorflow.org.

Agarwal, S., & Chowdary, C. R. (2021). Combating hate speech using an adaptive ensemble learning model with a case study on COVID-19. *Expert Systems with Applications*, *185*, Article 115632.

Aizawa, A. N. (2003). An information-theoretic perspective of tf-idf measures. *Inf. Process. Manag.*, *39*(1), 45–65. http://dx.doi.org/10.1016/S0306-4573(02)00021-3.

Aroyehun, S. T., & Gelbukh, A. F. (2018). 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@COLING, Santa Fe, New Mexico, USA, August 25* (pp. 90–97). Association for Computational Linguistics, URL https://aclanthology.org/W18-4411/.

Aroyo, L., Dixon, L., Thain, N., Redfield, O., & Rosen, R. (2019). Crowdsourcing subjective tasks: The case study of understanding toxicity in online discussions. In *Companion of the 2019 world wide web conference, WWW 2019, San Francisco, USA, May 13–17* (pp. 1100–1105). ACM, http://dx.doi.org/10.1145/3308560.3317083.

Bala, A., & Chana, I. (2015). Intelligent failure prediction models for scientific workflows. *Expert Systems with Applications*, *42*(3), 980–989. http://dx.doi.org/10.1016/j.eswa.2014.09.014.

Banerjee, S., Sarkar, M., Agrawal, N., Saha, P., & Das, M. (2021). Exploring transformer based models to identify hate speech and offensive content in english and indo-aryan languages. arXiv preprint arXiv:2111.13974.

Banerjee, S., Sarkar, M., Agrawal, N., Saha, P., & Das, M. (2021). Exploring Transformer Based Models to Identify Hate Speech and Offensive Content in English and Indo-Aryan Languages. In *Working notes (FIRE), Forum for information retrieval evaluation*. CEUR-WS.org.

Barbieri, F., Anke, L. E., & Camacho-Collados, J. (2021). XLM-T: a multilingual language model toolkit for Twitter. CoRR, abs/2104.12250. URL https://arxiv.org/abs/2104.12250. arXiv:2104.12250.

Basile, V., Bosco, C., Fersini, E., Nozza, D., Patti, V., Rangel Pardo, F. M., Rosso, P., & Sanguinetti, M. (2019). SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 54–63). Minneapolis, Minnesota, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S19-2007.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

Brown, A. (2020). *Models of governance of online hate speech*. Council of Europe, URL https://www.report-it.org.uk/files/models_of_governance_of_online_hate_speech_pdf.pdf.

Bunde, E. (2021). AI-assisted and explainable hate speech detection for social media moderators - A design science approach. In *54th Hawaii international conference on system sciences, HICSS 2021, Kauai, Hawaii, USA, January 5, 2021* (pp. 1–10). URL http://hdl.handle.net/10125/70766.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Strope, B., & Kurzweil, R. (2018). Universal sentence encoder for english. In *Proceedings conference on empirical methods in natural language processing, emnlp: system demonstrations, Brussels, Belgium, October 31–November 4* (pp. 169–174). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/d18-2029.

Chakravarthi, B. R., Priyadharshini, R., Muralidaran, V., Suryawanshi, S., Jose, N., Sherly, E., & McCrae, J. P. (2020). Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Forum for information retrieval evaluation* (pp. 21–24).

Chung, Y., Kuzmenko, E., Tekiroglu, S. S., & Guerini, M. (2019). CONAN - COunter NArratives through Nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th conference of the association for computational linguistics, ACL Florence, Italy, July 28–August 2* (pp. 2819–2829). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/p19-1271.

Davidson, T., Warmsley, D., Macy, M. W., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the eleventh international conference on web and social media, ICWSM 2017, Montréal, Québec, Canada, May 15–18, 2017* (pp. 512–515). AAAI Press, URL https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15665.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Long and short papers*: *Vol. 1*, *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N19-1423, URL https://aclanthology.org/N19-1423.

Dupont, D., Barbosa, J. L. V., & Alves, B. M. (2020). CHSPAM: a multi-domain model for sequential pattern discovery and monitoring in contexts histories. *Pattern Anal. Appl.*, *23*(2), 725–734. http://dx.doi.org/10.1007/s10044-019-00829-9.

Farooqi, Z. M., Ghosh, S., & Shah, R. R. (2021). Leveraging transformers for hate speech detection in conversational code-mixed tweets. arXiv preprint arXiv:2112.09986.

Filippetto, A. S., Lima, R., & Barbosa, J. L. V. (2021). A risk prediction model for software project management based on similarity analysis of context histories. *Information and Software Technology*, *131*, Article 106497. http://dx.doi.org/10.1016/j.infsof.2020.106497.

Fortuna, P., Soler Company, J., & Wanner, L. (2021). How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? *Inf. Process. Manag.*, *58*(3), Article 102524. http://dx.doi.org/10.1016/j.ipm.2021.102524.

Gorrell, G., Kochkina, E., Liakata, M., Aker, A., Zubiaga, A., Bontcheva, K., & Derczynski, L. (2019). SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 845–854). Minneapolis, Minnesota, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S19-2147.

Hande, A., Priyadharshini, R., & Chakravarthi, B. R. (2020). KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection. In *Proceedings of the third workshop on computational modeling of people's opinions, personality, and emotion's in social media* (pp. 54–63).

Indurthi, V., Syed, B., Shrivastava, M., Chakravartula, N., Gupta, M., & Varma, V. (2019). FERMI at SemEval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in Twitter. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 70–74). Minneapolis, Minnesota, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S19-2009.

Joshi, A., Prabhu, A., Shrivastava, M., & Varma, V. (2016). Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics*: *Technical papers*, (pp. 2482–2491).

Judge, M., & Nel, J. A. (2018). Psychology and hate speech: A critical and restorative encounter.

Kadam, A., Goel, A., Jain, J., Kalra, J. S., Subramanian, M., Reddy, M., Kodali, P., Arjun, T., Shrivastava, M., & Kumaraguru, P. (2021). Battling hateful content in Indic languages HASOC'21. arXiv preprint arXiv:2110.12780.

Kadam, A., Goel, A., Jain, J., Kalra, J. S., Subramanian, M., Reddy, M., Kodali, P., H, A. T., Shrivastava, M., & Kumaraguru, P. (2021). Battling Hateful Content in Indic Languages HASOC '21. In *Forum for information retrieval evaluation*: *Working notes (FIRE)*, CEUR-WS.org.

Khan, Y., Ma, W., & Vosoughi, S. (2021). Lone pine at SemEval-2021 task 5: Fine-grained detection of hate speech using BERToxic. CoRR, abs/2104.03506. URL https://arxiv.org/abs/2104.03506. arXiv:2104.03506.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Vol. 33*, *Advances in neural information processing systems* (pp. 18661–18673). Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1746–1751). Doha, Qatar: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/D14-1181.

Kim, Y. (2014). Convolutional neural networks for sentence classification. CoRR, abs/1408.5882. URL http://arxiv.org/abs/1408.5882. arXiv:1408.5882.

Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018). Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)* (pp. 1–11). Santa Fe, New Mexico, USA: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/W18-4401.

Lal, Y. K., Kumar, V., Dhar, M., Shrivastava, M., & Koehn, P. (2019). De-mixing sentiment from code-mixed text. In *Proceedings of the 57th annual meeting of the association for computational linguistics*: *Student research workshop*, (pp. 371–377).

Li, Q., Zhang, Q., & Si, L. (2019). EventAI at SemEval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 855–859). Minneapolis, Minnesota, USA: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S19-2148.

Madhu, H., Satapara, S., & Rathod, H. (2020). Astralis @ HASOC 2020: Analysis on identification of hate speech in indo-European languages with fine-tuned transformers. In P. Mehta, T. Mandl, P. Majumder, & M. Mitra (Eds.), *CEUR workshop proceedings*: *Vol. 2826*, *Working notes of FIRE 2020 - Forum for information retrieval evaluation, Hyderabad, India, December 16–20, 2020* (pp. 152–160). CEUR-WS.org, URL http://ceur-ws.org/Vol-2826/T2-7.pdf.

Mandl, T., Modha, S., Kumar M, A., & Chakravarthi, B. R. (2020). Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German. In *FIRE 2020*, *Forum for information retrieval evaluation* (pp. 29–32). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3441501.3441517.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandalia, C., & Patel, A. (2019). Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification inIndo-European languages. In P. Majumder, M. Mitra, S. Gangopadhyay, & P. Mehta (Eds.), *FIRE '19: Forum for information retrieval evaluation, Kolkata, India, December, 2019* (pp. 14–17). ACM, http://dx.doi.org/10.1145/3368567.3368584.

Mandl, T., & Womser-Hacker, C. (2005). A content independent model for context adaptation and individualization in information retrieval. In *Proceedings international workshop on context-based information retrieval (CIR-05) jointly with the 5th international and interdisciplinary conference on modeling and using context (CONTEXT-05) July 5, 2005 – Paris, France*. URL http://ceur-ws.org/Vol-151/CIR-05_5.pdf.

Menini, S., Aprosio, A. P., & Tonelli, S. (2020). A multimodal dataset of images and text to study abusive language. In *CEUR Workshop proceedings*: *Vol. 2769*, *Proceedings of the seventh italian conference on computational linguistics, CLiC-It 2020, Bologna, Italy, March 1–3, 2021*. CEUR-WS.org, URL http://ceur-ws.org/Vol-2769/paper_11.pdf.

Menini, S., Aprosio, A. P., & Tonelli, S. (2021). Abuse is contextual, what about nlp? The role of context in abusive language annotation and detection. CoRR, abs/2103.14916. URL https://arxiv.org/abs/2103.14916. arXiv:2103.14916.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Vol. 26*, In *Advances in neural information processing systems*. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Mikolov, T., & Zweig, G. (2012). Context dependent recurrent neural network language model. In *2012 IEEE spoken language technology workshop (SLT), Miami, FL, USA, December 2–5, 2012* (pp. 234–239). IEEE, http://dx.doi.org/10.1109/SLT.2012.6424228.

Modha, S., Majumder, P., & Mandl, T. (2018). Filtering aggression from the multilingual social media feed. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)* (pp. 199–207).

Modha, S., Majumder, P., & Mandl, T. (2021). An empirical evaluation of text representation schemes to filter the social media stream. *Journal of Experimental & Theoretical Artificial Intelligence*, 1–27. http://dx.doi.org/10.1080/0952813X.2021.1907792.

Modha, S., Majumder, P., & Mandl, T. (2022). An empirical evaluation of text representation schemes to filter the social media stream. *Journal of Experimental & Theoretical Artificial Intelligence*, *34*(3), 499–525. http://dx.doi.org/10.1080/0952813X.2021.1907792.

Modha, S., Majumder, P., Mandl, T., & Mandalia, C. (2020). Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance. *Expert Systems with Applications*, *161*, Article 113725. http://dx.doi.org/10.1016/j.eswa.2020.113725.

Modha, S., Mandl, T., Shahi, G. K., Madhu, H., Satapara, S., Ranasinghe, T., & Zampieri, M. (2021). Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo-Aryan languages and conversational hate speech. In *Forum for information retrieval evaluation* (pp. 1–3).

Modha, S., Mandl, T., Shahi, G. K., Madhu, H., Satapara, S., Ranasinghe, T., & Zampieri, M. (2021). Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo-Aryan languages and conversational hate speech. In D. Ganguly, S. Gangopadhyay, M. Mitra, & P. Majumder (Eds.), *FIRE 2021: Forum for information retrieval evaluation, virtual event, India, December 13–17, 2021* (pp. 1–3). ACM, http://dx.doi.org/10.1145/3503162.3503176.

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., & Cherry, C. (2016). SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th international workshop on semantic evaluation (SemEVal-2016)* (pp. 31–41). San Diego, California: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S16-1003.

Mundra, S., Singh, N., & Mittal, N. (2021). Fine-tune BERT to Classify Hate Speech in Hindi English Code-Mixed Text. In *Forum for information retrieval evaluation*: *Working notes (FIRE)*, CEUR-WS.org.

Parikh, P., Abburi, H., Chhaya, N., Gupta, M., & Varma, V. (2021). Categorizing sexism and misogyny through neural approaches. *ACM Transactions Web*, *15*(4), 17:1–17:31. http://dx.doi.org/10.1145/3457189.

Park, H., Cho, S., & Park, J. (2018). Word RNN as a baseline for sentence completion. In *5th IEEE international congress on information science and technology, CiSt 2018, Marrakech, Morocco, October 21–27* (pp. 183–187). IEEE, http://dx.doi.org/10.1109/CIST.2018.8596572.

Pavlopoulos, J., Sorensen, J., Dixon, L., Thain, N., & Androutsopoulos, I. (2020). Toxicity detection: Does context really matter? In *Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020, online, July 5–10, 2020* (pp. 4296–4305). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2020.acl-main.396.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pronoza, E., Panicheva, P., Koltsova, O., & Rosso, P. (2021). Detecting ethnicity-targeted hate speech in Russian social media texts. *Information Processing & Management*, *58*(6), Article 102674. http://dx.doi.org/10.1016/j.ipm.2021.102674.

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using siamese BERT-networks*. http://dx.doi.org/10.48550/ARXIV.1908.10084, URL https://arxiv.org/abs/1908.10084.

Ren, Y., Zhang, Y., Zhang, M., & Ji, D. (2016). Context-sensitive Twitter sentiment classification using neural network. In *Proceedings of the thirtieth AAAI conference on artificial intelligence, February 12–17, 2016, Phoenix, Arizona, USA* (pp. 215–221). AAAI Press.

Rosa, J. H., Barbosa, J. L. V., Kich, M., & Brito, L. (2015). A multi-temporal context-aware system for competences management. *Journal of Artificial Intelligence in Education*, *25*(4), 455–492. http://dx.doi.org/10.1007/s40593-015-0047-y.

da Rosa, J. H., Barbosa, J. L. V., & Ribeiro, G. D. (2016). ORACON: an adaptive model for context prediction. *Expert Systems with Applications*, *45*, 56–70. http://dx.doi.org/10.1016/j.eswa.2015.09.016.

Saha, K., Chandrasekharan, E., & De Choudhury, M. (2019). Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of the 10th ACM conference on web science* (pp. 255–264).

Salminen, J., Almerekhi, H., Kamel, A. M., Jung, S., & Jansen, B. J. (2019). Online hate ratings vary by extremes: A statistical analysis. In *Proceedings of the 2019 conference on human information interaction and retrieval, CHIIR 2019, Glasgow, Scotland, UK, March 10–14, 2019* (pp. 213–217). ACM, http://dx.doi.org/10.1145/3295750.3298954.

Sharma, M., Kandasamy, I., & Kandasamy, V. (2021). Deep learning for predicting neutralities in offensive language identification dataset. *Expert Systems with Applications*, *185*, Article 115458. http://dx.doi.org/10.1016/j.eswa.2021.115458.

Wang, W. Y. (2017). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th annual meeting of the association for computational linguistics, ACL Vancouver, Canada, July 30–August 4* (pp. 422–426). Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P17-2067.

Weston, J., Chopra, S., & Adams, K. (2014). #TagSpace: Semantic embeddings from hashtags. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1822–1827). Doha, Qatar: Association for Computational Linguistics, http://dx.doi.org/10.3115/v1/D14-1194.

Zaki, F., Sreyan, G., & Rajiv, S. (2021). Leveraging Transformers for Hate Speech Detection in Conversational Code-Mixed Tweets. In *Forum for information retrieval evaluation*: *Working notes (FIRE)*, CEUR-WS.org.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the type and target of offensive posts in social media. arXiv preprint arXiv:1902.09666.

Zarrella, G., & Marsh, A. (2016). MITRE at SemEval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th international workshop on semantic evaluation (SemEVal-2016)* (pp. 458–463). San Diego, California: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/S16-1074.