A Federated Approach to Predicting Emojis in Hindi Tweets

Anonymous ACL submission

Abstract

The use of emojis provide for adding a visual modality to textual communication. The task of predicting emojis however provides a challenge for computational approaches as emoji use tends to cluster into the frequently used and the rarely used emojis. Much of the research on emoji use has focused on high resource languages and conceptualised the task of predicting emojis around traditional servers-side machine learning approaches, which can introduce privacy concerns, as user data is transmitted to a central storage. We show that a privacy preserving approach, Federated Learning exhibits comparable performance to traditional servers-side transformer models. In this paper, we provide a benchmark dataset of 118k tweets (augmented from 25k unique tweets) for emoji prediction in Hindi and propose modification to the CausalFedGSD algorithm aiming to balance model performance and user privacy.\footnote{The dataset will be made publicly available upon request.} We show that our approach obtains comparative scores with more complex centralised models while reducing the amount of data required to optimise the models and minimising risks to user privacy.

1 Introduction

Since the creation of emojis around the turn of the millennium (Stark and Crawford, 2015; Al-shenqeeti, 2016), they have become of a staple of informal textual communication, expressing emotion and intent in written text (Barbieri et al., 2018b). This development in communication style has prompted research into emoji analysis and prediction for English (e.g. Barbieri et al., 2018a,b; Felbo et al., 2017; Tomihira et al., 2020; Zhang et al., 2020) while comparatively little attention has been given to the low resource languages.

Emoji-prediction has posed a challenge for the research community because emojis express multiple modalities, contain visual semantics and the ability to stand in place for words (Padilla López and Cap, 2017). The challenge is further compounded by the quantity of emojis sent and the imbalanced distribution of emoji use (Cappallo et al., 2018; Padilla López and Cap, 2017). Machine learning for emoji analysis and prediction has traditionally relied on traditional server-side architectures. However, training such models risk leaking sensitive information that may co-occur with emojis which can provide breaches of data privacy regulation (e.g. GDPR and CCPA). In contrast, federated learning (FL) (McMahan et al., 2017) approaches the task of training machine learning models by emphasising privacy of data. Such privacy is ensured by training models locally and sharing updates, rather than the data, with a central server (see Figure 1). The FL approach assumes that some client-updates may be corrupted during transmission. FL therefore aims to retain predictive performance while emphasising user privacy.

Motivated by prior work in privacy preserving machine learning (e.g. Ramaswamy et al., 2019; Yang et al., 2018) and emoji prediction for low resource languages (e.g. Choudhary et al., 2018b), we examine the application of FL to emoji prediction
for Hindi. Specifically, we collect an imbalanced dataset of 118,030 tweets in Hindi which contain 700 unique emojis that we classify into 10 predefined categories of emojis. We further examine the impact of two different data balancing strategies on federated and server-side, centralised model performance. Specifically, we examine: re-sampling and cost-sensitive re-weighting. The models under consideration are 6 centralised models that form our baselines: bi-directional LSTM (Hochreiter and Schmidhuber, 1997), IndicBert (Kakwani et al., 2020), HindiBERT, Hindi-Electra, mBERT (Devlin et al., 2019), and XLM-R (Conneau et al., 2020); and LSTMs trained using two FL algorithms: FedProx (Li et al., 2018) and a modified version of CausalFedGSD (Francis et al., 2021).

We show that LSTMs trained using FL perform competitively with more complex, centralised models in spite of only using up to 50% of the data.

2 Prior work

Federated learning Federated Learning (McMahan et al., 2017) is a training procedure which distributes training of models onto a number of client devices. Each client device locally computes weight updates on the basis of local data, and transmits the updated weights to the central server. In this way, FL can help prevent computational bottlenecks when training models on a large corpus while simultaneously preserving privacy by not transmitting raw data. This training approach has previously been applied for on-device token prediction on mobile phones for English. In a study of the quality of mobile keyboard suggestions, Yang et al. (2018) show that FL improves the quality of suggested words. Addressing emoji-prediction in English, Ramaswamy et al. (2019) use FL, to improve on traditional server-based models on user devices.

Centralised training In efforts to extend emoji prediction, Ma et al. (2020) experiment with a BERT-based model on a new English dataset that includes a large set of emojis for multi label prediction. Addressing the issue of low resource languages, Choudhary et al. (2018b) train a bi-directional LSTM-based siamese network, jointly training their model with high resource and low-resource languages. A number of studies on emoji prediction have been conducted in lower-resourced languages than English (e.g. Liebeskind and Liebeskind, 2019; Ronzano et al., 2018; Choudhary et al., 2018a; Barbieri et al., 2018a; Duarte et al., 2020; Tomihira et al., 2020). However, a commonality of these studies is the use of centralised machine learning models which compromise the privacy of users. Here, we study the use of FL for emoji prediction in low resource settings.

3 Data

We collect our dataset for emoji prediction by scraping ∼1M tweets using the Twitter API v2, keeping only the 24,794 tweets that contain at least one emoji and are written in Hindi. For tweets that contain multiple emojis, we duplicate the tweet by the number of emojis they contain and assign a single emoji to each copy, resulting in a dataset of 118,030 tweets with 700 unique emojis. Due to the highly imbalanced nature of emoji use in our dataset (see Figure 2), we categorise into a coarse-grained set of 10 emoji categories. Such simplifications, from multi-label to multi-class and unique emojis into emoji clusters risk losing semantic meaning that the emojis might hold. These choices however are motivated by how challenging the task of emoji prediction is, without such simplifications (Choudhary et al., 2018b).

3.1 Balancing data

This dataset exhibits a long-tail in the distribution of emoji categories (see Figure 3), with the vast majority of tweets belonging to the “Smileys & Emotions” and “People & Body” categories. To address this issue, we use two different data balancing methods: re-sampling (He and Garcia, 2009)

---

2These categories are obtained from the Emojis library, available at https://github.com/alexandrevicenzi/emojis.

3https://huggingface.co/monsoon-nlp/hindi-bert

4https://huggingface.co/monsoon-nlp/hindi-tpu-electra

---

Figure 2: Distribution of 15 most frequently appearing emojis in Hindi.
We conduct our experiments using PyTorch (Paszke et al., 2019) and Transformers (Wolf et al., 2020) on Google Colab using a Nvidia Tesla V100 GPU with 26GB of RAM. The datasets are split into train (80%), validation (10%), and test sets (10%). We measure our performance using precision, recall, and weighted F1. Each model is trained and evaluated on the original imbalanced data and two balancing approaches (see Section 3.1). Finally, for the federated setting, we conduct experiments where data is independent and identically distributed (I.I.D.) across the different client nodes.

### 4.1 Baseline models

We use 6 centralised models as baselines to compare the federated approach against. Specifically, we use a bi-LSTM (Hochreiter and Schmidhuber, 1997) with 2 hidden layers and dropout at 0.5, two multi-lingual models: mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). Finally we use IndicBERT (Kakwani et al., 2020), HindiBERT, and Hindi-Electra as these are pre-trained on Indic languages. All baselines are trained with batch size 8, learning rate $4e^{-5}$, and seq. length 128.

### 4.2 Federated models

For our federated experiments, we use the FedProx (Li et al., 2018) and a modified version of the CausalFedGSD (Francis et al., 2021) algorithms. FedProx trains models by considering the dissimilarity among the local gradients and uses a proximal term to the loss function to prevent divergence when the data is not I.I.D.

CausalFedGSD trains models by sharing a global subset of raw data with all local clients, where local and global data are concatenated to compute the weight updates. We modify CausalFedGSD such that the global model is initialised on 30% of the data and subsequently all weight updates are computed locally.

We reuse the Bi-LSTM (see Section 4.1) as our experimental model on client devices due to its relative low requirements for compute. For our experiments, we set the number of clients to 100 and simulate I.I.D. and non-I.I.D. settings. We simulate an I.I.D. setting by ensuring that all client devices receive data that is representative of the entire dataset. For the non-I.I.D. setting, we create severely imbalanced data splits for clients by first grouping the data by label, then splitting the grouped data into 200 bins and randomly assigning 2 bins to each client. We experiment with three different settings, in which we randomly select

---

6IndicBERT is pre-trained on 12 Indic languages, HindiBERT and Hindi-Electra are both trained on Hindi Wikipedia and CommonCrawl.

7We set the value of the proximal term to 0.01 following Li et al. (2018).
Table 1: Centralised model performances.

<table>
<thead>
<tr>
<th>c</th>
<th>Bi-LSTM</th>
<th>mBERT</th>
<th>XLM-R</th>
<th>IndicBERT</th>
<th>hindiBERT</th>
<th>Hindi-Electra</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>50%</td>
<td>68.74</td>
<td>92.99</td>
<td>59.48</td>
<td>68.04</td>
<td>62.44</td>
<td>64.58</td>
</tr>
<tr>
<td>30%</td>
<td>6.64</td>
<td>61.82</td>
<td>68.74</td>
<td>62.95</td>
<td>55.16</td>
<td>57.92</td>
</tr>
</tbody>
</table>

Table 2: Results using the FedProx algorithm. c is the percentage of clients whose updates are considered.

<table>
<thead>
<tr>
<th>c</th>
<th>Bi-LSTM</th>
<th>mBERT</th>
<th>XLM-R</th>
<th>IndicBERT</th>
<th>hindiBERT</th>
<th>Hindi-Electra</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>50%</td>
<td>87.04</td>
<td>63.68</td>
<td>82.04</td>
<td>81.56</td>
<td>61.82</td>
<td>64.58</td>
</tr>
<tr>
<td>30%</td>
<td>63.72</td>
<td>62.55</td>
<td>63.72</td>
<td>61.79</td>
<td>57.76</td>
<td>54.66</td>
</tr>
</tbody>
</table>

Table 3: Results using the modified CausalFedGSD. c is the percentage of clients whose updates are considered.

Table 4: F1-scores for the best performing centralised and federated models.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Centralised</th>
<th>Federated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi-LSTM</td>
<td>FedProx</td>
<td>Modified CausalFedGSD</td>
</tr>
<tr>
<td>Imbalanced</td>
<td>63.83</td>
<td>63.60</td>
</tr>
<tr>
<td>Re-sampled</td>
<td>58.61</td>
<td>52.14</td>
</tr>
<tr>
<td>Cost-Sensitive</td>
<td>64.46</td>
<td>63.78</td>
</tr>
</tbody>
</table>

4.3 Analysis

Considering the results for our baseline models (see Table 1), we find that XLM-R and IndicBERT obtain the best performances. We find that using a cost-sensitive weighting tends to out-perform re-sampling the dataset. Specifically, we find that the cost-sensitive weighting performs comparably with other settings or out-performs them. Curiously, we find that Hindi Electra under-performs compared to all other models, including HindiBERT which is a smaller model trained on the same data. This discrepancy in the performances of these two models may be due to the differences in complexity, and thus data required to achieve competitive performances. Finally, the bi-LSTM slightly under-performs in comparison to XLM-R, however it obtains competitive performances with all other well-performing models.

Turning to the performance of the federated baselines (see Table 2), we find an expected performance of the models. Generally, we find that the federated models achieve comparative performances, that are slightly lower than the centralised systems. Considering the F1-scores, we find that the optimal setting of the ratio of clients is subject to the data being I.I.D. In contrast, models trained on the re-sampled data tend to prefer data in an I.I.D. setting, but in general under-perform in comparison with other weighting strategies, including the imbalanced sample. Using our modification of the CausalFedGSD algorithm, we show improvements over our FL baselines when the data is I.I.D. and variable performance for a non-I.I.D. setting (see Table 3). Comparing the performances of the best performing settings, we find that the FL architectures perform comparably with the centralised models, in spite of being exposed to less data and preserving privacy of users (see Table 4).

5 Conclusion

Emoji prediction in user-generated text is a task which entails potentially highly private data, hence it is important to consider privacy-preserving methods for the task. Here, we presented a new dataset for emoji for Hindi and compared a privacy preserving approach, Federated Learning, with the centralised server-trained method and also a modified approach to the CausalFedGSD algorithm (Francis et al., 2021) to perform federated learning. Experimenting with the different data balancing methods and simulating settings where data is I.I.D. and non-I.I.D, we find that using federated learning can afford comparable performances to the more complex fine-tuned language models trained centrally, while ensuring privacy. In future work, we plan to extend this work to multi-label emoji prediction and investigate strategies for dealing with decay of the model vocabulary.
Ethical considerations

The primary reason for using federated learning is to ensure user-privacy. The approach can then stand in conflict with open and reproducible science, in terms of data sharing. We address this issue by making our dataset open to the public, given that researchers who are at institutions without IRB processes, data will only be released given a research statement that also details potential harms to participants.

Our modification of the CausalFedGSD model introduces the concern of some data being used to initialise the model. Here a concern can be that some data will be available globally. While this concern is justified, the use of federated learning affords two things: First, federated learning can limit on the overall amount of raw data that is transmitted and risks exposure. Second, initialisation can occur using synthetic data, created for the express purposes of model initialisation. Moreover, pre-existing public, or privately owned, datasets can be used to initialise models, which can be further trained given weight updates provided by the client nodes. Federated learning, and our approach to federated learning thus reduce the risks of exposing sensitive information about users, although the method does not completely remove such risks.

References


A Appendix

A.1 Data

The tweets were curated using the "Elevated access" to the Twitter API v2. Using a developer account, we query tweets written in Hindi language that are up to 512 characters long. Multiple occurrences of tweets due to re-tweeting were discarded. Figure 4 shows a sample of tweets present in our Hindi dataset for the task of emoji prediction.

A.2 Server-Based Models

For traditional server-side transformer models, we use the simpletransformers\textsuperscript{10} library. We use the default configuration options. We train all the transformer models for 25 epochs with a learning rate of 4e-5 and no weight decay or momentum.

A.3 Federated Learning Plots

This section provides detailed graphs comparing the training loss, validation AUC, validation F1 score and validation accuracy for every dataset variation. All of these graphs were made using Weights and Biases (Biewald, 2020).

\textsuperscript{10}https://simpletransformers.ai/
<table>
<thead>
<tr>
<th>Lang</th>
<th>Text</th>
<th>Emoji</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi</td>
<td>तुझे कितना चाहने लगे हम</td>
<td>🌸</td>
<td>Food &amp; Drink</td>
</tr>
<tr>
<td>English</td>
<td>How much we love you</td>
<td>👨‍❤️‍👨</td>
<td>People &amp; Body</td>
</tr>
<tr>
<td>Hindi</td>
<td>एकदम जबरदस्त भावनात्मक मीठास शब्दों के सरसरी हवाओं में कुछ अजीब सी खुशबू।</td>
<td>🌹</td>
<td>Animals &amp; Nature</td>
</tr>
<tr>
<td>English</td>
<td>Some strange fragrance in the whispering winds of very emotional sweet words</td>
<td>🍫</td>
<td>Objects</td>
</tr>
<tr>
<td>Hindi</td>
<td>जन्मिदन की अनंत शुभकामनायें</td>
<td>🎂</td>
<td>Smiley &amp; Emotion</td>
</tr>
<tr>
<td>English</td>
<td>Best wishes for your birthday</td>
<td>🎉</td>
<td>Smiley &amp; Emotion</td>
</tr>
</tbody>
</table>

Figure 4: Example of our Hindi dataset

A.3.3 Balanced Dataset (IID)  

A.3.4 Balanced Dataset (non-IID)
A.3.5 Cost Sensitive Approach (IID)

Chapter 3

A.3.6 Cost Sensitive Approach (non-IID)

A.4 Time vs GPU Usage

This section provides detailed graphs for GPU usage in Watts for every variation of experiments run.
A.4.1 Imbalanced Dataset

A.4.2 Balanced Dataset

A.4.3 Cost Sensitive Approach