Federated Learning with Diversified Preference for Humor Recognition

Xu Guo[1] , **Pengwei Xing**[1] , **Siwei Feng**[1] , **Boyang Li** [1] , **Chunyan Miao**[1,2]

[1]School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
[2]Alibaba-NTU Singapore Joint Research Institute
{xu008, pengwei.xing, siwei.feng, lily.liboyang, ascymiao}@ntu.edu.sg

Abstract

Understanding humor is critical to creative language modeling with many applications in human-AI interaction. However, the perception of humor is personal due to different humor preferences. Thus, a given passage can be regarded as funny to different degrees by different readers. This makes training humorous text recognition models that can adapt to diverse humor preferences highly challenging. In this paper, we propose the FedHumor approach to recognize humorous text contents in a personalized manner through federated learning (FL). It is a federated BERT model capable of jointly considering the overall distribution of humor scores with humor labels by individuals for given texts. Extensive experiments demonstrate significant advantages of FedHumor in recognizing humor contents accurately for people with diverse humor preferences compared to 9 state-of-the-art humor recognition approaches.

1 Introduction

Humor plays an important role in social communications. Unlike many objective classification tasks, the task of humor recognition is constrained by its subjectivity. Given the same jokes, it is difficult to achieve consensus among people due to the fact that human preferences lead to different degrees of perception on what is funny [Aykan and Nalçacı, 2018], which is as illustrated in Figure 1. This makes it challenging for humor recognition models to generalize to more users in practice, as they are trained on humorous and non-humorous examples labelled by experimenters [Yang *et al.*, 2015; Chen and Soo, 2018] or employed annotators [Zhang and Liu, 2014; Hossain *et al.*, 2019]. To achieve more robust humor recognition from a wider range of populations, it is necessary to consider diverse preferences of users.

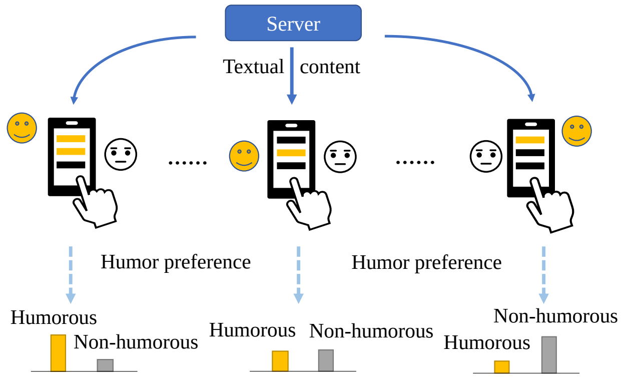

Figure 1: User preferences lead to disagreement on what is funny.

Previous research on automated humor recognition cast the task as a binary classification problem [Yang *et al.*, 2015; Chen and Soo, 2018]. These methods mainly focus on how to design humor-related linguistic features as input to a classifier to obtain high classification performance. With well-established computational humor theories [Shultz, 1976; Gruner, 1997; Binsted *et al.*, 2006], they can curate many heuristics to extract informative features. The key of heuristic rules is to design effective approaches to capture special linguistic patterns [Yang *et al.*, 2015; Liu *et al.*, 2018] or to count n-gram statistics [Taylor and Mazlack, 2004] that can distinguish humorous text from plain text. These methods are able to characterize intra-sentence and/or inter-sentence dependencies that are unique to humor, and thus do not rely a lot on the complexity of classifiers. Nevertheless, the feature generation process requires significant efforts and many have difficulties to cope with newly encountered terms [Zhang and Liu, 2014].

Deep learning has shed light on the feature engineering using neural networks. Especially, Convolutional Neural Networks [Chen and Soo, 2018] and Transformer-based language models [Mao and Liu, 2019] has been used for end-to-end humor recognition. Most of previous studies are conducted on curated datasets where their humorous and non-humorous examples are taken from different sources [Yang *et al.*, 2015; Liu *et al.*, 2018; Chen and Soo, 2018] or different distributions [Zhang and Liu, 2014; Mao and Liu, 2019] and then prepared to be balanced. This experimental setting implies an underlying assumption that people all agree what are treated as humorous and what are non-humorous, which neglected human preferences and thus limited their applicability in practice.

Federated Learning, a technique that trains a deep neural network based on iterative averaging of decentralized local updates, has been proved to be robust to unbalanced and non-IID data distributions [McMahan *et al.*, 2017]. Inspired by recent progress of federated learning in diversity [Ramaswamy *et al.*, 2019] and personalization [Arivazhagan *et al.*, 2019], we propose to improve the generalization ability of humor recognition models on a variety of user preferences with the help of federated learning. We name the model as - FedHumor. Especially, we adopted the Federated Averaging (FedAvg) algorithm [McMahan *et al.*, 2017] as the base training setting, then fine-tuned a pretrained Transformer-based language model on our task, and employed a diversification strategy [Ramaswamy *et al.*, 2019] to handle diverse user preferences.

The main idea of our solution is to force the humor recognition model to learn from a diverse range of user preferences, thereby enhancing the adaptability to new users. For this purpose, there are two important issues to consider. First, as users are increasingly aware of privacy issues and reluctant to provide personal information [Yang *et al.*, 2019], it is impossible to aggregate explicit user preference information that reside on their personal devices. To address this, we propose an approximation strategy to generate implicit user feedback (i.e., labels) on given humorous text and we diversify the label distributions to represent diverse user preferences. Second, marginal distributions of user preferences (reflected as hardly amused or easily amused personalities) often lead to salient class imbalance issue which requires us to select a more suitable evaluation metric rather than widely adopted accuracy. As such, we use F1 score, the harmonic mean of precision and recall, to evaluate and select best models.

To the best of our knowledge, FedHumor is the first federated learning-based humor recognition model. Extensive results show that our approach is able to increase the generalization bounds of the humor recognition model compared to 9 state-of-the-art approaches. It is a promising approach to help future AI applications recommend suitable humorous texts to users under a stricter data privacy protection landscape [Yang *et al.*, 2019], thereby enabling more complex human-AI interactions to emerge.

2 Related Work

The mainstream verbal humor recognition approach leverages a set of humor-related features from linguistic perspectives to train a text classification model that distinguishes humorous text from non-humorous text. Traditional humor studies use static or stereotyped instances of humor found in literary compilations or online repositories. [Mihalcea and Strapparava, 2005] retrieved a large number of one-liners from web pages using explicitly indicated humor-related contents. They designed three humor-specific stylistic features: alliteration, antonym, and adult slang and trained a classifier to detect humorous one-liners.

Statistical analysis on humor-specific language structures has also been leveraged for humorous text recognition. [Yang *et al.*, 2015] extracted a set of statistical features from four humor-related semantic structures: incongruity, ambiguity, interpersonal effect and phonetic style to trained a Random Forest classifier based on a combination of these handcrafted features under Word2Vec representations. With the development of social platforms, new forms of expression emerge. [Zhang and Liu, 2014] collected a set of tweets and let a native English speaker to filter out humorous ones. They combined humor-related linguistic features suited for Twitter texts to train a Gradient Boosted Regression Trees classifier to detect humorous tweets.

To avoid manually designing curated features, deep neural networks have been applied to learn humor-related features automatically from humorous texts. [Chen and Lee, 2017] designed a Convolutional Neural Network (CNN) to recognize verbal humor from TED talks. [Chen and Soo, 2018] augmented CNN with the Highway Network [Srivastava *et al.*, 2015] to improve performance in short joke detection. [Mao and Liu, 2019] utilized the pretrained multilingual BERT and fine-tuned it on the HAHA task [Chiruzzo *et al.*, 2019] to recognize humorous tweets in Spanish.

Although deep learning-based humorous text recognition models do not rely on feature engineering, the labels in the training datasets they use depends on the annotators' personal preference. Such biased datasets may limit the applicability of the trained model on other user populations. Recent FL text processing models start to focus on personalization to cope with statistical heterogeneity among different participants' local datasets [Arivazhagan *et al.*, 2019; Ramaswamy *et al.*, 2019]. However, they assume that different users face different data instances. In this work, we address the personalization issue of the same texts viewed by different users producing diverse humor labels.

3 Methodology

In this section, we illustrate our problem context and present the detailed design of FedHumor solution for humor recognition.

3.1 Preliminaries

Task definition. A traditional humor recognition task is defined as: given a piece of text X, a humor recognition model, denoted as f, should predict it as humorous or not: $f(X) = y, y \in \{0, 1\}$, where the label y is empirically determined by a small scope of annotators. Our task extends the setting by relaxing the confidence of the labels, where the model f should predict the label that are agreed by the majority of n different user preferences: $f(x) = \mathrm{Maj}(y_i, ..., y_n)$. While such information can not be determined directly due to privacy protection, it is iteratively learned through federated learning.

Implicit label generation. Jokes should have different intensities of funniness and different people will give different ratings to the same jokes [Hossain *et al.*, 2017]. However, it is difficult to aggregate such explicit feedback from a diverse population on the same jokes in the privacy protection setting. As such, we approximately generate implicit labels and diversify their distributions, which heavily rely on two assumptions: (1) Given a set of n jokes $\mathcal{S} = \{(X_1, \alpha_1), ..., (X_n, \alpha_n)\}$ sorted by their degrees of funniness, $\alpha \in A | A = (\alpha_1, ...\alpha_n)$, people would agree on the jokes whose funniness is close to the marginal boundaries of

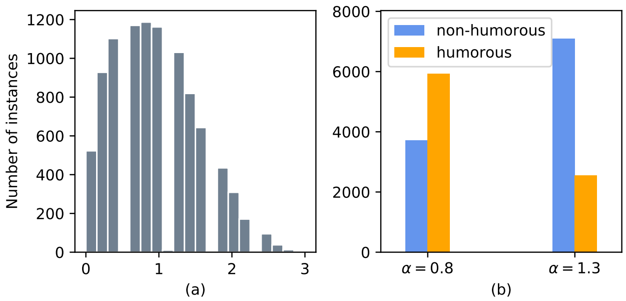

Figure 2: Implicit label generation strategy. (a) The distribution of explicit funniness ratings on a set of jokes. (b) Implicit humorous and non-humorous label distributions from different user preferences controlled by the user preference α.

the funniness interval, $\max_{\forall s \in \mathcal{S}} \alpha(s)$ and $\min_{\forall s \in \mathcal{S}} v(s)$ (i.e., not funny and very funny), and may disagree on the middle jokes that are far away the two extremes (e.g., slightly funny and moderately funny). (2) A user's preference is characterized by a specific transition point in the funniness interval $\alpha_k \in A$ where the user would dislike the jokes below the funniness value α_k and like the jokes beyond that funniness. In this paper, we only consider scenarios that satisfy the two assumptions to investigate effects of federated learning on the generalization capability of humor recognition models. Based on the assumptions, the liked and disliked results are treated as implicit humorous and non-humorous labels, $y \in \{0, 1\}$, for the humor recognition models to learn. To gain an overview, the resulted implicit label distributions that imply users' different personal preferences are illustrated in Figure 2.

3.2 Humor Recognition through Federated Learning

The purpose of federated learning solution for humor recognition is to enhance the generalization performance of traditional humor recognition models when dealing with new user preferences under the privacy protection policy. In order to learn such a model without aggregating personal preference information into a central storage location, FedHumor is designed similar to Google's GBoard scenario [Ramaswamy *et al.*, 2019] based on the FL paradigm. We employed the Transformer-based encoder - BERT [Devlin *et al.*, 2018] to capture contextualized sentence representations as the feature learning part of our model and fine-tune its pretrained weights on our task. BERT tokenizes the jokes into tokens in its vocabulary and encodes every sequence of tokens $T = (t_1, ..., t_m)$ into 12 distinct hidden states, $\{\boldsymbol{H}_i\}_{i=1}^{12} | \boldsymbol{H}_i = (\boldsymbol{h}_1, ...\boldsymbol{h}_m)$, which are computed in parallel through 12 self-attention layers, where $\boldsymbol{H} \in \mathbb{R}^{r_h \times m}$ and $r_h = 768$ in our task. A linear layer, as a classifier to be optimized, is applied on the pooled hidden states followed by Softmax operation to predict the labels of jokes. Users join the federated learning following the same protocol as in Federated Averaging [McMahan *et al.*, 2017].

Federated Model Training

Robustness is a major concern in federated learning as decentralized devices may contain corrupted data, such as diverse preferences on jokes in our task, making the federated learning system vulnerable to arbitrarily skewed distributions [Ghosh *et al.*, 2019]. To mitigate this shortcoming, we incorporated an adaptation mechanism into the federated training which aims to personalize model predictions to local distributions. The mechanism can provide punishment or reward and the degree is determined by the empirical distributions on each client device. As shown in Eq. (1), the predicted probability of each class $\widehat{P}(y_i = y)$ for i-th joke X_i is *scaled* by dividing them over the empirical probability of each class $P(y_i = y)$ for a specific user, where $y \in \{0, 1\}$ and $P \in (0, 1)$ as we assume users will perceive at least some of the jokes as humorous and some are not.

$$\widetilde{P}(y_i = y|X_i) = \text{Softmax}\left(\frac{\widehat{P}(y_i = y|X_i)}{P(y_i = y)^{\beta}}\right) \quad (1)$$

where β is a scaling factor which can be determined empirically through experiments and $\text{Softmax}(x_i) = exp(x_i) / \sum_j exp(x_j)$ aims to recompute the adjusted probabilities. The denominator term $P(y_i = y)^{\beta}$ overall serves as either a punishment or a reward depending on the comparison of predictions and actual distributions. For example, if the predicted probability is $\widehat{P}(y_i = 1) = 0.9$ while the actual distribution for $y = 1$ is $\widehat{P}(y_i = 1) = 0.7$, then the denominator serves as a punishment to adjust the prediction to be $\widetilde{P}(y_i = 1) \approx 0.72$ for $\beta = 1$, which is closer to the actual distribution. Such empirical probabilities are determined locally according to each client's personal preference α.

We follow the FedAvg algorithm to aggregate client model parameter updates after each round of local training to produce a global model. At each training round t, the current global FL model with parameters w_t is sent to k clients randomly selected from the device population. A more detailed discussion about the selection strategy can be found in [McMahan *et al.*, 2017]. Each client is initialized with a unique personal humor preference α to produce implicit labels, as introduced in Section 3.1. Eq. 2 shows the optimization problem on client $m, \forall m \in [1, k]$:

$$\min_{w_t^{(m)}} -\frac{1}{N} \sum_{i=0}^{N} \sum_{y=0}^{1} \log(\widetilde{P}(y_i = y|X_i, w_t^{(m)})) \quad (2)$$

where N is the number of instances in a minibatch and $\widetilde{P}(y_i = y)$ is the scaled probability for each prediction as defined in Eq. (1). The client weight parameter updates are then averaged across devices for the FL server to compute the new global FL model parameters, $w_{t+1} = \frac{1}{k} \sum_{m=1}^{k} w_t^{(m)}$. The FL model training process of FedHumor is illustrated in Figure 3. Throughout the process, User' sensitive personal humor preference information did not leave their own device.

Federated Inference

After completing FL model training, the global FL model will be transferred to the clients to facilitate personalized humor

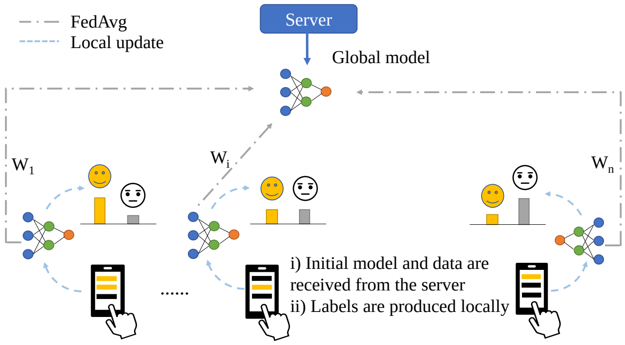
FedAvg
Local update
Server
Global model
W1
Wi
Wn
......
i) Initial model and data are received from the server
ii) Labels are produced locally

Figure 3: The training process of FedHumor: 1) the FL server sends the current global model and text contents to a group of clients; 2) the clients provide humor labels for the contents based on their own preferences, update the model locally, and send their model parameters to the server; 3) the server aggregates these local updates to produce a new global model.

recognition. In this stage, the predicted probability $\widehat{P}(y_i = y)$ for i-th joke X_i on each client is *re-scaled* by multiplying it with the empirical probability $P(y_i = y)$ stored on that client, as shown in Eq. (2).

$$\widetilde{P}(y_i = y|X_i) = \mathrm{Softmax}(\widehat{P}(y_i = y|X_i)P(y_i = y)^{\beta}) \tag{3}$$

The scaling and re-scaling operations are performed on each client locally. By applying Eq. (1) in the FL model training stage and Eq. (2) in the federated inference stage, we achieve the goals of learning a federated model for humorous text recognition from decentralized data sources with diverse preference, and customizing the federated model when it is applied for inference by each client locally. The pseudo-code for FedHumor is given in Algorithm 1.

Algorithm 1: FedHumor. The K clients are indexed by k; D_{tr} is the training set and D_{te} is the test set; β set is set by the FL server; α_k depends on client k; B is the number of batches; E is the number of local epoches; and η is the learning rate.

```
Server executes:
    Initialize $w_0$ from pretrained weights;
    for each round $t = 1, 2, ..., T$ do
        for each client $k = 1, 2, ..., K$ in parallel do
            $w_{t+1}^k \leftarrow \mathrm{ClientUpdate}\ (k, w_t, \beta)$;
        $w_{t+1} \leftarrow \frac{1}{K}\sum_{k=1}^{K} w_{t+1}^k$;

ClientUpdate (k,w,$\beta$) : // run on client k
    Compute prior probability
      $P(y_k = 0|\alpha_k), P(y_k = 1|\alpha_k)$;
    Select $\beta_k$ from $\beta$ based on $\alpha_k$;
    $\mathcal{B} \leftarrow$ (split $D_{tr}$ into batches)
    for each local epoch $i = 1, 2, ..., E$ do
        for batch $b \in \mathcal{B}$ do
            Compute loss $\mathcal{L}$ based on Eq. (1) and Eq.
              (2);
            $w \leftarrow w - \eta \bigtriangledown \mathcal{L}(w; b)$;
    return $w$ to server

Inference:
    for each client $k = 1, 2, ..., K$ do
        $\mathcal{B} \leftarrow$ (split $D_{te}$ into B batches)
        Given $P(y_k = 0|\alpha_k), P(y_k = 1|\alpha_k), \beta_k$;
        for batch $b \in \mathcal{B}$ do
            Make predictions based on Eq. (3);
        Evaluate global model performance on client $k$;
```

4 Experimental Evaluation

In this section, we study the properties of FedHumor and evaluate its performance against 9 state-of-the-art humor recognition approaches through experiments on real-world data.

4.1 Real-world Dataset

To study humor recognition on a diverse populations, we use a challenging dataset from SemEval-2020 Task 7 - assessing the funniness of edited news headlines [Hossain *et al.*, 2019], which contains humorous text that are edited using incongruous words. The training/dev/test sets consist of 9,652/2,419/3,024 instances and they provide fine-grained funniness scores to every edited news headline, with intensity ranging from 0 to 3. For example, a news headline titled - *Royal wedding : Meghan's dress in detail* - was micro-edited by replacing *dress* with *elbow* to produce a funny version - *Royal wedding : Meghan's elbow in detail* - which received an average funniness score of 2 by five selected annotators.

We reuse the dataset as a basis to experimentally study the effectiveness of FedHumor for handling diverse personal humor preferences. We generated implicit labels and diversify the distributions for different users using the strategy introduced in Section 3.1. Given the same jokes, increasing the threshold of being amused or what we call user preference (α) can make more samples being annotated as non-humorous, as show in Figure 4.

4.2 Evaluation Setup

We set up the evaluation metric for comparing our federated learning based humor recognition model with other comparable models, and for selecting the best models in total training epochs. Other experimental settings were set as default.

Generalizability evaluation. The generalization capability of a humor recognition model can be measured by its performance on unseen data set that has different marginal distribution from the observed set. Intuitively, the higher the test performance it produces, the more generalizable a model is to new users. A relevant proof of generalization bound versus robustness is discussed in [Xu and Mannor, 2012].

Metric selection. Diverse preferences of humor can result in very unbalanced datasets stored on different users. Under this condition, a model can achieve an artificially high accuracy by predicting all the labels to be the dominant class. In order to provide fair evaluation for model performance, we consider *precision*, *recall* and *F1-score* in instead of the widely used accuracy metric. Macro average is more strict than other averaging methods when computing overall performance on an imbalanced dataset. For example, if a model

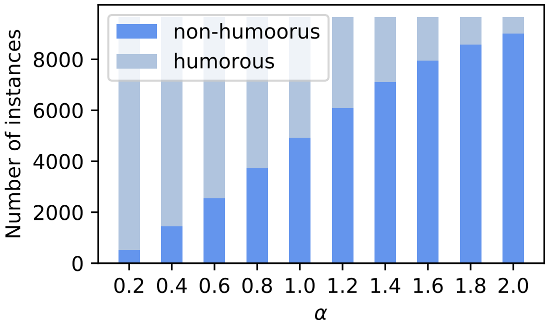

Figure 4: Implicit humorous/non-humorous label distributions in the dataset after personal humor preferences are taken into account.

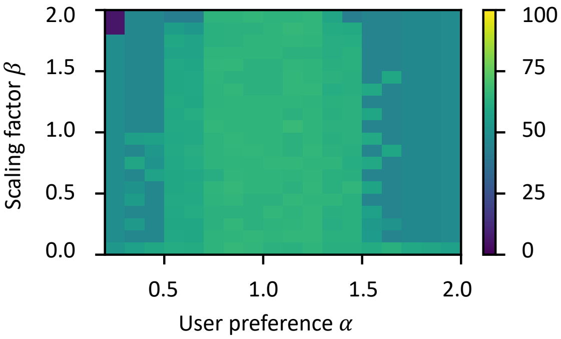

Figure 5: Sensitivity analysis for parameter α and β on model performance evaluated using F1 score.

achieves 100% recall on a negative class with 90 samples and 0% recall on a positive class with 10 samples, macro averaging produces an overall recall of 50% while micro averaging produces an overall recall of 90%. Therefore, we use macro averaging to report the precision, recall and F1-score achieved by each model on the test dataset in order to pay more attention to their performance on the minority class.

We adopt the following metrics to evaluate the performance of a model based on the i-th user preference:

1. Macro Precision: $P_i = \frac{1}{2}(\frac{TP}{TP+FP} + \frac{TN}{TN+FN})$
2. Macro Recall: $R_i = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$
3. Macro F1: $F_i = \frac{TP}{2 \cdot TP+FN+FP} + \frac{TN}{2 \cdot TN+FN+FP}$

Where TP, TN, FP and FN are calculated based on how many humorous instances are classified as True Positive (TP) or False Negative (FN) and how many non-humorous instances are classified as True Negative (TN) or False Positive (FP).

To evaluate the performance of a model on a diverse range of n users, we adopt the following metrics:

1. Overall Precision: $P = \frac{1}{n} \sum_{i=1}^{n} P_i$
2. Overall Recall: $R = \frac{1}{n} \sum_{i=1}^{n} R_i$
3. Overall F1: $F = \frac{1}{n} \sum_{i=1}^{n} F_i$

4.3 Experiment 1: Sensitivity Analysis

In this experiment, we aim to understand the impact of diverse humor preferences (α) on the performance of FedHumor and to what extent personalization (controlled by β) should be applied. To do so, we varied the values of α from low degree ($\alpha = 0.2$) to high degree ($\alpha = 2.0$), and at the same time, increased the scale factor β from small value ($\beta = 0$) to large value($\beta = 2.0$), all in increments of 0.1, to conduct a number of tests. We regenerate the humor labels each time α is varied. We trained the model for each combination of α and β and the best model is selected using F1 score.

The results on test set are shown in Figure 5. For every α value, there are multiple β values that can achieve the best personalization performance. It means that our model is not very sensitive to β and a reasonable value would be suffice. In general, a model can achieve better performance on a preference range of $(0.5, 1.5)$, which produce a non-humorous label distribution ranging from 20% to 75%. For users who have too low ($\alpha < 0.5$) or too high ($\alpha > 1.5$) humor preference values (which results in a very unbalanced dataset), β is recommended to set to a low value (e.g., around 0.1). For users who have medium humor preference value, the model performance is not sensitive to value of β, making it robust under such conditions.

4.4 Experiment 2: Federated Learning

We evaluate federated learning based approach with other comparable learning strategies for humor recognition on two groups of clients with different degrees of diversity:

- Group 1: a small group of 3 clients with humor preference values $\alpha = [0.3, 0.9, 1.8]$, representing an easily amused personality, a neutral personality and a hardly amused personality; and
- Group 2: a large group of 18 clients with humor preference values, α, ranging from 0.2 to 1.9 in an increment of 0.1, representing a more diverse range of population.

For both groups, we empirically set the value of scaling factor β to 0.1 for clients whose preferences α are below 0.5 or above 1.5 and set β to 1.0 for those clients whose preference values are within the range $[0.5, 1.5]$. They are determined through iterative experiments as discussed in Section 4.3.

We design two intuitive learning strategies to train a humor recognition model and compare them with the one using federated learning strategy, described as follows:

1. **AGG**: to aggregate all the implicit labels from local client into the central database, and treat them as ground truth to learn a BERT-based humor recognition model, which follows the traditional way of training machine learning models.
2. **INDV**: to train an individual humor recognition model for every user based on their local ground truth labels. The same penalty mechanism as the one in FedHumor is applied on every user to cope with their individual class imbalance problems. However, this solution is not practical when we want to scale up to a large population of users. In addition, whenever a new user comes, a new model needs to be established.

Table 1: Results of three learning strategies on the test set.

		Precision	Recall	F1
Group 1	AGG	58.59	54.89	41.66
	INDV	56.30	55.32	53.52
	FED	**60.03**	**65.57**	**55.61**
Group 2	AGG	57.40	51.25	33.05
	INDV	58.14	55.61	53.03
	FED	**61.67**	**66.62**	**57.48**

3. **FED**: as we introduced in Section 3, this approach is the federated learning-based humor recognition model incorporated with a distribution adaptation mechanism.

The average performance of the approaches on each group on their test sets are shown in Table 1. FED achieves better performance in terms of precision, recall and F1-score than AGG and INDV on both the small group and the large group. FED is better than INDV means the issue of data insufficiency is alleviated by federated learning, which is the core of FedHumor. FED is better than AGG means FedHumor considers personalization and can be adapted to different users.

4.5 Experiment 3: Humor Recognition

The advent of BERT [Devlin *et al.*, 2018] has opened a new era in natural language research by presenting SOTA results on many downstream NLP tasks. These models are pretrained on large amounts of existing text using self-supervision with no data annotation required, and thus are often treated as encoders primed with knowledge of a language. In this experiment, we compare the performance of FedHumor against the following humor recognition baseline models and BERT variants on the personalized humorous text recognition task.

- **DV-LR**: We trained Doc2Vec [Mikolov *et al.*, 2013] using the distributed bag of words approach and applied logistic regression for classification.
- **WV-RF**: We reproduced the humor recognition model in [Yang *et al.*, 2015], which used a pretrained Word2Vec model for sentence representation and a Random Forest for classification.
- **WV-CNN-HN**: We reproduced the model in [Chen and Soo, 2018], which augmented a CNN with a Highway Network for humor recognition.
- **BERT-FZ**: We applied the pretrained BERT model and freezed the model parameters. A fully connected layer is applied on top of BERT and is trained for classification.
- **BERT-FT**: We applied the pretrained BERT and fine-tuned its parameters together with a classification layer.
- **BERT-L/C/M-FT**[1]: We fine-tuned other versions of BERT, including the one pretrained on a much larger dataset (BERT-L), the one pretrained with words cased (BERT-C), and the one pretrained on multilingual dataset (BERT-M), and model parameters are updated together with the classification layer to form three baseline approaches.

Table 2: Results for Experiment 3 (in percentage).

	Precision	Recall	F1
DV-LR	53.69	53.67	53.64
WV-RF	56.70	56.10	55.20
WV-CNN-HN	56.20	54.70	51.90
BERT-FZ	54.15	53.71	52.53
BERT-FT	64.91	64.88	64.87
BERT-L-FT	64.48	64.48	64.47
BERT-C-FT	62.69	62.65	62.62
BERT-M-FT	62.11	62.08	62.06
ALBERT-FT	61.06	61.05	61.04
FedHumor	**66.60**	**66.56**	**66.53**

- **ALBERT-FT**: We fine-tuned one of the upgraded BERT variant - ALBERT [Lan *et al.*, 2019] on our humor recognition task for comparison.
- **FedHumor**: The proposed approach which pretrained the BERT model under FL framework on a group of 18 diverse humor preferences. We apply the model on the same test set to compare with the other humor recognition models.

User preference value α is fixed at 1 in this experiment, making its test dataset balanced (as presented in Figure 4). All the comparison models are trained using the same training set and FedHumor is trained following its own paradigm. Their performance on the same test set is reported in Table 2.

The first three models (i.e., DV-LR, WV-RF, WV-CNN-HN) are stationary word representation models which are widely adopted in humor recognition tasks before the advent of transformers. Their performance are generally not good on this dataset. The following six models (i.e., BERT-FZ, BERT-FT, BERT-L/C/M-FT, ALBERT-FT and FedHumor) are BERT variants. The results of BERT-FZ are much worse than the rest, which shows that fine-tuning pretrained models can enable better feature representations. FedHumor, which is powered by FL diagram, achieved better results than the above models across metrics. This can be attributed to the federated training and personalization strategy that forced the model to learn from diversified user preferences and thus presented better generalization performance on the test set.

5 Conclusions

In this paper, we propose the FedHumor approach - a humorous text recognition model following the federated learning paradigm which can provide personalized humor recognition based on labels stored in distributed sources. It is able to account for diversity in each person's activation point for perceived funniness for the same text contents. Through extensive experiments comparing FedHumor with 9 state-of-the-art approaches, we show that it is able to achieve better personalization when recognizing humor from text contents. To the best of our knowledge, it is the first federated learning-based personalized humorous text recognition model.

[1] https://huggingface.co/transformers/pretrained_models.html

Acknowledgments

This work is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore.

References

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *CoRR, arXiv:1912.00818*, 2019.

[Aykan and Nalçacı, 2018] Simge Aykan and Erhan Nalçacı. Assessing theory of mind by humor: The humor comprehension and appreciation test (tom-hcat). *Frontiers in psychology*, 9, 2018.

[Binsted *et al.*, 2006] Kim Binsted, Anton Nijholt, Oliviero Stock, Carlo Strapparava, G Ritchie, R Manurung, H Pain, Annalu Waller, and D O'Mara. Computational humor. *IEEE Intelligent Systems*, 21(2):59–69, 2006.

[Chen and Lee, 2017] Lei Chen and Chong MIn Lee. Convolutional neural network for humor recognition. *CoRR, arXiv:1702.02584*, 2017.

[Chen and Soo, 2018] Peng-Yu Chen and Von-Wun Soo. Humor recognition using deep learning. In *NAACL*, pages 113–117, 2018.

[Chiruzzo *et al.*, 2019] Luis Chiruzzo, S Castro, Mathias Etcheverry, Diego Garat, Juan José Prada, and Aiala Rosá. Overview of haha at iberlef 2019: Humor analysis based on human annotation. In *IberLEF*, 2019.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR, arXiv:1810.04805*, 2018.

[Ghosh *et al.*, 2019] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

[Gruner, 1997] Charles R. Gruner. The game of humor: A comprehensive theory of why we laugh. *Transaction Publishers*, 1997.

[Hossain *et al.*, 2017] Nabil Hossain, John Krumm, Lucy Vanderwende, Eric Horvitz, and Henry Kautz. Filling the blanks (hint: plural noun) for mad libs humor. In *EMNLP*, pages 638–647, 2017.

[Hossain *et al.*, 2019] Nabil Hossain, John Krumm, and Michael Gamon. "President Vows to Cut Hair": Dataset and analysis of creative text editing for humorous headlines. *CoRR, arXiv:1906.00274*, 2019.

[Lan *et al.*, 2019] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *CoRR, arXiv:1909.11942*, 2019.

[Liu *et al.*, 2018] Lizhen Liu, Donghai Zhang, and Wei Song. Exploiting syntactic structures for humor recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1875–1883, 2018.

[Mao and Liu, 2019] Jihang Mao and Wanli Liu. A bert-based approach for automatic humor detection and scoring. In *IberLEF*, 2019.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, pages 1273–1282, 2017.

[Mihalcea and Strapparava, 2005] Rada Mihalcea and Carlo Strapparava. Making computers laugh: Investigations in automatic humor recognition. In *EMNLP*, pages 531–538, 2005.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013.

[Ramaswamy *et al.*, 2019] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *CoRR, arXiv:1906.04329*, 2019.

[Shultz, 1976] Thomas R Shultz. A cognitive-developmental analysis of humour. 1976.

[Srivastava *et al.*, 2015] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *NeurIPS*, pages 2377–2385, 2015.

[Taylor and Mazlack, 2004] Julia M Taylor and Lawrence J Mazlack. Computationally recognizing wordplay in jokes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 26, 2004.

[Xu and Mannor, 2012] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

[Yang *et al.*, 2015] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. Humor recognition and humor anchor extraction. In *EMNLP*, pages 2367–2376, 2015.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning*. Morgan & Claypool Publishers, 2019.

[Zhang and Liu, 2014] Renxian Zhang and Naishi Liu. Recognizing humor on twitter. In *CIKM*, pages 889–898, 2014.