Visual Analysis of Heterogeneous Text Data based on Federated Learning

Yufan Zhou, Jiayin Li, Dingding Chen, Chengtao Ji,Yirui Huang, Chao Wu, Lingyun Yu

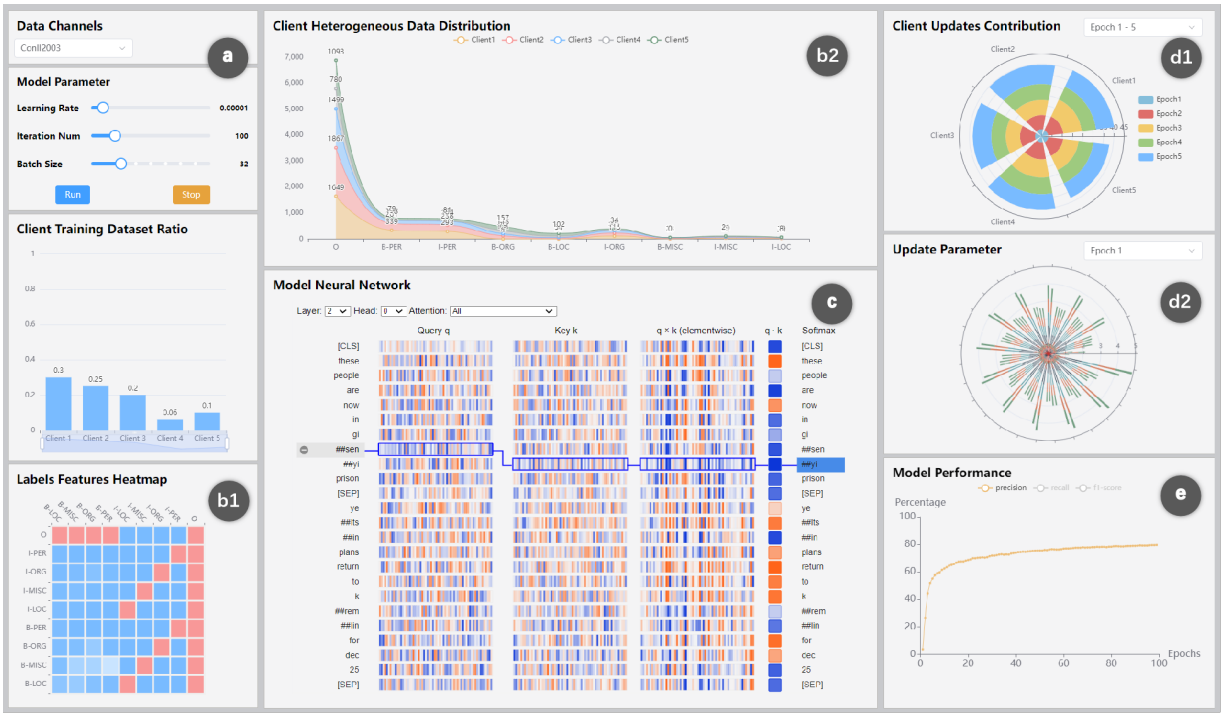

Figure 1: The main interface for analyzing the heterogeneous text data via federated learning cooperation. (a) The hyperparameter adjustment component allows users to fine-tune hyperparameters interactively. (b) The heterogeneous data detection component provides visual monitoring of heterogeneous text data in federated learning, enabling quality evaluation of server-side and client-side datasets, and detecting data using images as references. (c) The model neural network component offers a neural network visualization with server-side test set data, facilitating model inference interpretation. (d) The global model update components monitor the parameter updates from clients to the server, evaluate each client's contributions to the global model updates, and assess the impact of clients on specific model parameters. (e) The model performance evaluation component is updated in real-time during the federated learning training process, displaying the performance metrics of the global model in testing.

Abstract

In recent years, with the rise of distributed computing and increasing emphasis on privacy by people, NLP Federated Learning has rapidly emerged. As a distributed deep learning model, federated learning allows users to use local data for model training while protecting privacy. However, the heterogeneity of data in federated learning presents significant challenges for model training, such as device unavailability or difficulty in model convergence. Moreover, the privacy-preserved nature of federated learning makes it challenging for researchers to observe data distribution clearly and adjust the model accordingly. To address these issues, we developed a visualization tool aimed at helping users explore the impact of heterogeneous text data so that researchers can design more efficient models. In our solution, we show researchers the distribution of data between clients and servers, as well as the state of the neural network during the training process. To allow researchers to tune models more quickly, we also provide options for adjusting deep learning parameters, selecting multiple datasets, and intuitively displaying training results. Our tool can support users to design more realistic and optimized models. The effectiveness of the visualization tool was validated by an expert review.

Keywords: Heterogeneous data, visualization, federated learning, natural language processing

1 Introduction

With the development of the Internet, an increasing number of mobile devices are being used, generating vast amounts of data that hold tremendous value [17]. These data are always distributed among different owners, restricted by laws and privacy policies, making it difficult to aggregate or directly share them among different regions or organizations [11]. Federated learning–a distributed model training approach that conducts on users' devices and aggregates the updated weights on a central server–has emerged as a promising solution [15], as it can effectively address the problem of information silos while preserving user privacy. Especially in Vertical Federated Learning, users only need to share their respective trained model weights information of their own features instead of sharing raw data. This approach is particularly suitable in scenarios where data is sensitive and homogenous data types. Given its advantages, vertical federated learning has been widely applied in fields such as healthcare [29], finance [14], government administration [6], and natural language processing [30].

However, one of the most critical challenges of vertical federated learning is data heterogeneity. Not identically and independently distributed data (Non-IID) can lead to weight divergence in local models [8, 33], resulting in increased complexity of modeling and theoretical analysis [9]. Additionally, it can make the model difficult to converge and significantly reduce accuracy [3]. Processing natural language processing (NLP) data in federated learning poses a significant challenge due to the diversity of natural language expressions, language characteristics, and domain knowledge across various application fields. This makes solving the data heterogeneity problem in NLP data a challenging task. For example, if a client's data contains more specialized vocabulary than other clients, the model may overemphasize these unique features, leading to poor overall data generalization. On the other hand, clients with insufficient training data may train inadequate models, which could adversely affect upstream collaborations with other functionalities.

Various efforts have been made to address the challenge of data heterogeneity in federated learning, primarily focusing on two complementary aspects: improving the efficiency of model aggregation and limiting the bias caused by local models to the global model [16]. However, these approaches have mostly been validated only on image datasets, and there is still a limited amount of research and application in the field of natural language processing [10], dispite its importance in various domains. Furthermore, the non-visibility of data in federated learning makes it difficult to adjust the model in the presence of Non-IID data, further adding to the complexity of the task.

In this paper, we propose a visual analytic approach to assist users in analyzing the impact of heterogeneous data in NLP federated learning. Our aim is to help users explore various aspects of NLP federated learning under Non-IID conditions, including data distribution, feature selection, neural network dynamics, and model parameter optimization. The contributions of this paper include the following:

- We develop a novel visual analysis tool to assist in analyzing the impact of heterogeneous data in NLP-FL models.
- A dynamic interface is provided to help users understand the changes of neural networks and parameters during the federated learning process.
- We assist users in gaining a visual understanding of NLP-FL models and designing better federated learning models.

2 Related Work

In this section, we review relevant work on NLP in federation learning, heterogeneous data in federation learning, as well as on visual model analysis.

2.1 NLP in Federated Learning

NLP data is often scattered among many different data holders and contains extremely sensitive information [29]. Federated Learning (FL), a decentralized machine learning approach, can aggregate data from distributed sources without centralizing the data and effectively address issues such as data shafts, data privacy, and data security [34]. Therefore, researchers continue to explore the application of FL in NLP tasks to meet the growing demand for natural language data analysis.

Currently, the application of FL in NLP tasks has been successful in many fields, such as sentiment analysis [18], text classification [1], and machine translation [19]. In terms of platform design, Lin et al. designed a benchmark framework [10] for evaluating federation learning performance on various tasks and implemented a common interface between Transformer-based language models and federation learning methods under various non-IID partitioning strategies. Cai et al. proposed the FedAdapter framework [2] to improve the model convergence speed of NLP-FL.

However, NLP-FL also faces challenges due to the inherent nature of natural language, such as data imbalance caused by distributional differences across devices, making it difficult to learn a model that generalizes well. The parameter size of the pre-trained model is also an issue, which can make it challenging to converge during federated learning [12]. To address these challenges, we aim to make each step of the federation learning training process transparent to users. This will help them understand how data distribution and parameter changes impact the model, ultimately improving modeling efficiency.

2.2 Heterogeneity in Federated Learning

Heterogeneity in federation learning arises from many sources, including heterogeneous feature spaces, unbalanced data distributions, unstable network connections, and limited device resources. The main approaches to cope with data imbalance focus on three aspects, such as designing new loss functions, performing data expansion, and using data sampling [4, 5, 21].

In this paper, we focus on addressing and showing users the heterogeneity caused by unbalanced data distribution. The academic community is currently focusing on three aspects of data sampling, data expansion, and designing new loss functions to cope with data imbalance [11].

Improvements to the cross-entropy loss function [20] or the use of inverse distance aggregation [31] can to some extent alleviate the performance degradation caused by data imbalance. However, these methods for designing loss functions have limitations, as they only show significant effects on some tasks and datasets. Undersampling most of the types according to the data distribution of different samples is also one of the solutions, however, this approach leads to an insufficient number of certain categories and thus makes it difficult to classify these categories accurately [28]. Hao and Zhang et al. verified the effectiveness of data augmentation in overcoming the effects of heterogeneous data for federal learning from different perspectives [7, 32].

However, these methods have only been validated on partial image datasets or are only effective in classification tasks. It is more important for FL to use visualization methods to enable users to better understand heterogeneous problems and thus design more generalized models.

2.3 Visual Model Analysis

The visual model analysis in machine learning is divided into three main categories: monitoring model performance fluctuations, checking model configuration, and input and output analysis [26].

In the existing applications, the FATEBoard component under the FATE framework can visually display contents such as logs and evaluation results to monitor model performance fluctuations, however, it is not able to provide detailed fault analysis on the client side [27]. HetVis [26] derives the presence of heterogeneous data by observing anomalies in the training process and allows users to observe the distribution and check for dissimilar clusters. Li et al. designed a visualization tool HFLens [9] specifically for longitudinal federated learning (HVL), which allows simple inspection of model performance for correlation analysis, and potential anomaly checking. However, the model is customized for FATE, it may not be readily applicable to longitudinal federated learning more broadly.

To check the details of the model configuration, GANViz [25] observes the impact of a feature on the global model by comparing image features, while DGMTracker [13] locates the neurons that are causing the model to fail in training. Jesse Vig designed a complete set of visualization tools [22, 24] that allow users to enter sentences on their own and observe the changes in the bert or transformer neural network during training.

3 Design Considerations

In this section, we illustrate the design aspects of the visualization system, including the target users, the requirements analysis, and the overall architecture of the system.

3.1 Target Users and Tasks

The system is designed for researchers and enterprises working in the field of FL, with interactive visualization tools that facilitate their research and work by presenting complex datasets, models, and parameters. Specifically, the proposed visualization system can be particularly useful for researchers in academia and industry who are interested in FL, machine learning, and natural language processing. By utilizing visual tools to explore and analyze client data distribution, datasets' features, and other factors related to FL in NLP, researchers can better understand the impact of parameters and fine-tune the model for optimal performance.

One of the key benefits of the proposed visualization techniques is the ability to compare the global and local models in FL. This allows researchers to observe how different clients contribute to the overall performance of the global model, and identify any potential bottlenecks or imbalances in the training process. Furthermore, by visualizing the data distribution and model parameters for each client, researchers can gain insights into the factors that affect model performance, and make informed decisions on how to adjust the model to improve the accuracy. The visualization tools can also provide a framework for monitoring the training process and tracking changes in the model over time, helping researchers to evaluate the effectiveness of different training strategies and optimize the model for specific NLP tasks.

3.2 Requirements Analysis

The process of collecting requirements for the FL visualization system involved several discussions with experts in the related field. These discussions were conducted in various ways, including face-to-face meetings, phone calls, and email exchanges. The experts included researchers, practitioners, and industry professionals who have experience working with FL and/or NLP. During the discussions, we asked the experts to share their opinions and insights on a range of topics related to the visualization of FL and NLP. Specifically, we sought their feedback on the current challenges and opportunities in the field, the potential benefits and limitations of different visualization techniques, and the features that would be most critical in a successful visualization system. By gathering input from these experts, we were able to identify several key requirements for the visualization system:

R1. The Impact of Data Heterogeneity: To successfully train an FL model, it is critical to account for the heterogeneity of data across devices. Independent training of models on each device results in data being saved solely on the local devices, with only model parameters sent to the server for aggregation. Therefore, the global model's performance relies on the local data distribution's similarity across all devices. Understanding and identifying how heterogeneous data affects the global model and the FL process can significantly benefit the users.

R1.1 Detecting Non-IID data. FL faces significant challenges due to the Non-IID nature of data distribution across participating clients. This Non-IID property leads to the performance degradation of trained models. Visualizations can provide intuitive visual analysis of the data and identify clients with Non-IID data, which can help to mitigate the negative effects of Non-IID. Researchers can use these visualizations to better understand how Non-IID data is distributed across the clients and detect clients with anomalous or biased data to resolve the problem in FL.

R1.2 Observing the parameter updates. The global model on the server side is iteratively updated based on the parameter received from the participating clients. However, due to data heterogeneity, these parameter updates returned by clients may have significant differences or even conflicts. Visualizing the parameter updates can assist users in quickly identifying the differences in parameter updates between clients and detecting clients with conflicting updates. Quantifying and visualizing these parameter updates would allow researchers to understand and account for the variability in training data across the participating clients. It can also inform decision-making on how to measure the contributions from each client's updates appropriately, facilitating accurate communication and aggregation of model updates across the federation of devices.

R2. Model Performance: Model performance is of critical importance to users, as it directly affects the quality of the final model. Due to the distributed nature of FL, the performance requirements for such models are often high, as they have to perform accurately on each user's local data while also providing good performance on the aggregated data. As such, users may need to evaluate the performance of models using various benchmarking techniques to ensure that the resulting models are both accurate and comprehensive.

R2.1 Dynamic displaying model performance. The system should allow users to receive real-time feedback on model performance and identify issues as they arise. By providing constant updates on key performance metrics, dynamic displaying can help users make informed decisions about their models and streamline the training process.

R2.2 Demonstrating the neural network layers. Visualizing neural network layers provides valuable insights into the performance and internal workings of machine learning models in FL. It can help users understand which layers are most significant in driving model behavior and identify potential performance issues or biases. By visualizing these layers, users can also gain deeper insight into how their data is being transformed and processed through the neural network. Additionally, layer visualization can be used to identify and understand the impact of any changes made to the model architecture on its overall performance.

R3. Interactivity: For a successful and effective Federated Learning process, users require visual analysis systems that possess strong interactivity capabilities and can seamlessly interact with data and backend FL modules.

R3.1 Allowing interacting with the generated charts. Users can manipulate visualizations dynamically based on their needs and preferences, improving the overall understanding of the data. Interactive charts offer a high level of flexibility and customization, enabling users to explore visual representations of complex information more intuitively.

R3.2 Allowing hyperparameter adjustments. Based on the feedback from the majority of users we interviewed, the ability to adjust FL hyperparameters through an interactive interface is highly desirable and in high demand. FL involves multiple parties sharing model updates without sharing raw data, which can lead to complex optimization challenges that require tuning of hyperparameters. By allowing users to interactively adjust hyperparameters in real-time, they can train the models easily with various settings, optimize their models faster, and ultimately achieve improved performance.

4 Heterogeneous Text Data Visualization

In this section, we will systematically describe the entire heterogeneous text data visualization system and its components.

4.1 System Overview

Our system is structured into three main modules:

- A simulation module for vertical FL, which could simulate the vertical FL process in a real-world scenario.
- A data collection and visualization module that generates visualizations based on the data generated during the FL process.
- A web-based user interface that presents the visualizations and allows users to interactively adjust the hyperparameters of the FL module.

The proposed simulation module facilitates the training of models using vertical FL techniques. For implementing **R1** requirement, the availability of both global model data on the server side and local model data on each client is necessary. Though in a real-world FL scenario, accessing specific dataset information and local model information on the server side is not feasible due to user privacy concerns. However, given the requirements of local data for **R1**, our FL simulation module is data-transparent and allows the use of data generated throughout the FL process. Our proposed strategy entails leveraging server-provided datasets and allocating subsets datasets as training material to individual clients during the simulation process within an FL scenario. The assembly of client-specific training data forms the foundation of our data analysis and visualization process, which serves as a key evaluation metric for assessing both the effectiveness and efficiency of the FL process and satisfies the **R2** requirement. Moreover, we utilize the update parameters returned by clients to examine the impact of local models on the global model, as proposed in **R1**.

The data collection and visualization module plays a critical role in collecting and visualizing key statistics related to the FL process, including information on the dataset, model, and performance of the system. By generating visualizations of these statistics, the module satisfies the requirements outlined in the requirement analysis (**R1**, **R2**), providing users with important insights about the behavior of the FL algorithm and any abnormal data that might be present. The generated visualizations demonstrate important aspects of the FL process, such as model accuracy, convergence rates, and potential data drift. These insights help users assess the overall quality of the dataset, identify any sources of bias or inadequacies, and evaluate the effectiveness of the FL approach in achieving meaningful results.

The web-based user interface serves as a crucial entry point for users to interact with the FL system. Through a user-friendly and intuitive design, the interface displays meaningful visualizations generated by the data collection and visualization module, providing users with valuable insights into the performance and behavior of the system. Furthermore, the user interface satisfies the requirement specified in the requirement analysis (**R3**), allowing users to adjust the hyperparameters of the FL module directly through an interactive interface. Users can observe the effects of hyperparameter adjustments on the generated visualizations, enabling them to evaluate the effectiveness of different hyperparameter configurations for achieving optimal performance.

These three system modules collectively constitute the framework of our comprehensive FL visual analysis system. This system enables a complete process from simulating vertical FL to information visualization, meeting all user requirements collected during the requirements analysis phase. By leveraging each of these modules, our system extracts critical insights from the FL process, such as data quality and model performance metrics, enabling users to evaluate the effectiveness of the overall system at each step of the learning process. Through this analysis, our system provides valuable feedback to help optimize the performance of the system over time, ultimately leading to better outcomes and predictions.

4.2 Vertical Federated Learning

In order to simulate realistic vertical FL scenarios, we developed a user interface that allowed for flexible hyperparameter adjustments (Fig. 2), such as modifying the learning rate, iteration number, batch size, and training dataset ratio for individual clients. Once users had

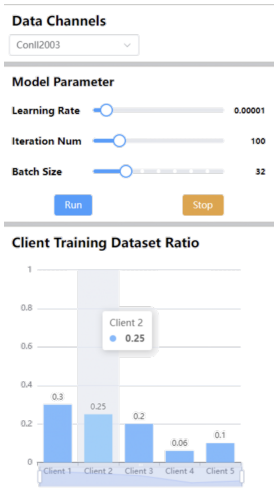

Figure 2: The hyperparameters control component

made their hyperparameters selections, these hyperparameters were confirmed and sent to the backend.

To accurately track the impact of heterogeneous data during the FL process, we implemented a transparent and interpretable architecture. Each client was assigned a sub-training set by dividing the entire dataset into subsets based on user-specified training set ratios. At the end of each global epoch, the local clients would communicate the updated parameter values to the server, which would then update or generate new visualizations via a visualization module. This approach facilitated transparency, interpretability, and interactivity throughout the FL process. Our FL simulation allowed us to evaluate the performance of vertical FL in realistic settings while providing valuable insights into the impact of data heterogeneity on learning outcomes.

4.3 Datasets Features

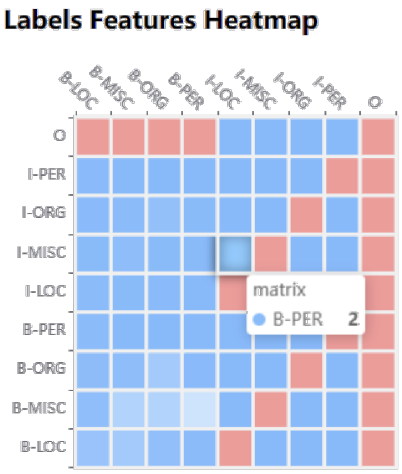

Figure 3: Labels features heatmap

Prior to conducting FL, we preprocessed the training dataset by transforming it into a feature representation with Named Entity Recognition (NER) labels. NER is a critical natural language processing task with numerous applications, including information extraction and machine translation. These NER labels served as crucial features for FL in our simulated scenario.

Our visualization system includes a dedicated component for NER analysis that examines transitions between adjacent named entity labels. This feature generates a heatmap that demonstrates the frequency of label transitions, allowing for a better understanding of the classification process. Fig. 3 shows the heatmap, wherein the x and y-axes represent the entity labels, and color intensity represents the frequency of transitions between labels as the darker red the higher frequency of transitions between two NER labels. The heatmap facilitates the identification of frequently occurring label pairs and highlights areas that may require further improvement. A custom color gradient was also used to highlight label pairs with higher frequencies, enhancing the accuracy of the visualization. The chosen data encoding method, a heatmap, was selected for its ability to illustrate the transition patterns between entity labels effectively.

The choice of adjacent NER tag conversion frequency as the feature measure is based on two main reasons:

a) The NER adjacent tag conversion frequency is particularly valuable for handling heterogeneous text data. This type of data often originates from various domains, languages, and text types, each with unique entity characteristics and patterns. Consequently, NER models need to be trained and evaluated on different data types. By using the adjacent label conversion frequency as a proxy, it becomes possible to guide the training and application of NER models across diverse texts, spanning various domains, languages, and text types.

b) The frequency of adjacent label transitions can assess the NER model's performance. A good NER model should adhere to the actual language's entity occurrence rules, where the adjacent label transition frequency should be aligned with the patterns of entity occurrences in the language. If the NER model's adjacent label transition frequency does not match the language's pattern, further model tuning or dataset quality improvement may be needed. Understanding and analyzing transition patterns between entity labels are vital for optimizing the model's performance. By gaining insights into the model's decision-making process for adjacent labels using a heatmap, we can identify areas for improvement and guide further optimization of the model's performance.

4.4 Client Heterogeneous Text Data Distribution

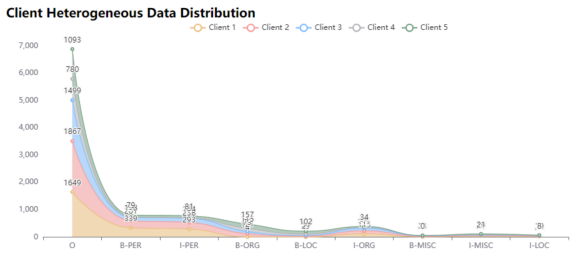

Figure 4: Client heterogeneous data distribution statistics chart

In our simulated FL scenario, the client training data is a subset of the server dataset sliced based on the hyperparameters provided by the user. Therefore, the dataset is fully transparent and interpretable in this simulation. We utilize a client heterogeneous data distribution statistics chart (Figure 4) to display the distribution of textual data labels among different clients, assisting users in observing the heterogeneous text data on each client. Through visualization, patterns can be easily identified and insights obtained to assist NLP model development or optimization. Furthermore, identifying the distribution of named entity labels in different clients can help detect any anomalies or errors in the data, which is crucial in FL settings, to address label imbalance in clients and ensure that the model can generalize well to new datasets.

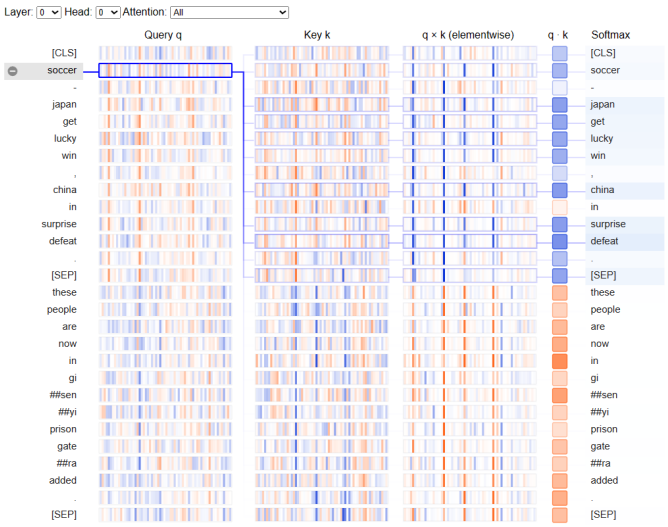

Figure 5: Model neural network visualization

4.5 Neural Network Display

Neural network visualization is a critical aspect of deep learning research as it helps researchers gain insight into the behavior of specific layers and feature propagation between them. This type of visualization provides users with an intuitive understanding of how a model performs classification and similarity calculations. Moreover, it helps in optimizing the model and explaining its decision-making process.

In our visualization system, we used the BERT model for NLP tasks in the process of FL. To visualize the neural networks, we chose Bertvis, which is an excellent tool for exploring the internal workings of BERT models [23]. The data for BertVis visualization in the study were selected based on the similarity calculation of the test set used on the server side.

For the NER task, we measure the degree of similarity between data by *cosine similarity*:

$$\text{sim}(a,b) = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \times \sqrt{\sum_{i=1}^{n} (b_i)^2}} \quad (1)$$

Here, a and b are the two vectors being compared, n is the length of the vectors, and a_i and b_i represent the i-th element of vectors a and b, respectively.

Cosine similarity is a widely-used method for measuring the similarity between two vectors. In the context of NLP, it is common for different forms of the same named entity or co-referenced entities to appear in different sentences. By applying cosine similarity to the vectors representing the NER-related content of two sentences, we can identify semantically related sentences despite the presence of different entities or entity forms.

In the system's data collection and visualization module, each sentence is a vector, with each element of the vector representing a specific NER label in the dataset. The value of each element reflects the frequency of the corresponding NER label. The cosine similarity measure is calculated for these vectors to estimate the similarity between the NER-related content of two sentences. By extracting the maximum similarity among all sentence pairs, we can identify the most similar pair of sentences in terms of NER-related content. The advantage of cosine similarity is its ability to consider NER label frequency in each sentence and to account for sentence variation in length and structure, making it well-suited to this task.

The method of selecting data for a visualization based on similarity calculation enables the BertVis tool to concentrate on the most pertinent and characteristic data for analysis and interpretation (Fig. 7 5). The cosine similarity metric has been used effectively to capture the semantic similarities between data points while considering the grouping of similar NER labels. This approach can be readily modified to suit other NLP tasks and datasets for the purpose of data selection and visualization.

It generates a BERT neural network visualization that comprises input sequences, tokens, and attention matrices using the most similar pair of sentences in terms of NER-related content. The crucial aspect of the visualization is the attention matrix, which quantifies the relationship strength between different input tokens. To interpret the attention scores, one considers Query (Q), Key (K), and the attention operation. Q is the token being analyzed, and K represents other tokens. The element-wise product of Q and K (q x k) calculate pairwise similarity, and the softmax of q x k yields attention scores indicating the importance of Key for predicting Query. Users analyze the q x k matrix for the significant attention scores, indicating highly relevant contextual information, and compare it to other sequences for identifying regularities and anomalies of interest. By visualizing the attention scores and patterns, users can identify the significant features of the data that contribute to the model's performance and compare them across different clients. This allows for better insights into the variance in the data and enables the model to aggregate knowledge effectively.

4.6 Model Updates Contribution

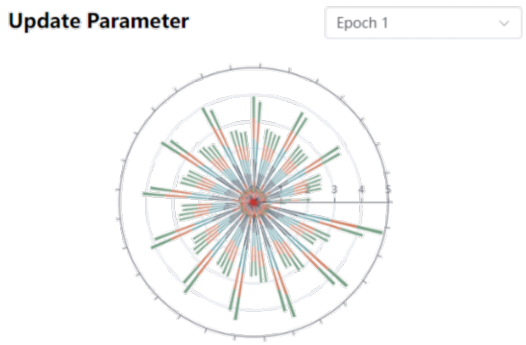

Figure 6: Client updates contribution polar coordinate chart

The model updates monitor consists of two visualization components: a) one for monitoring the contribution of client parameter updates in each epoch (Figure 6); b) one for observing the specific updates of each update parameter returned by each client to the global model in each epoch round (Figure 7).

a) The client contribution visualization is a component designed to visualize client parameter updates during the FL process. When the global model is updated after each FL round, the visualization module receives a nested list containing epoch values and the different parameter updates of each client for each epoch as input. This module iterates through the client parameter updates of each epoch, computes their *L2 norms*, and aggregates them to evaluate the overall client contribution. Finally, this aggregated *L2 norm* value is visualized using a polar coordinate chart that stacks the contribution of each client for each epoch.

$$|x|2 = \sqrt{\sum i = 1^n x_i^2} \tag{2}$$

Here, $|x|_2$ represents the L2 norm of vector x, x_i represents the i^{th} element of vector x, and n is the length of vector x.

The L2 norm is a mathematical concept that is commonly used as a measure of the magnitude or the length of a vector. In the context of FL, the client parameter updates can be seen as vectors, and their L2 norm represents the magnitude of the update in a particular direction in the parameter space. By calculating the L2 norm of the client parameter updates, we can evaluate the amount of the update and compare it with other updates in the same epoch. Moreover, taking the sum of L2 norms across all clients during an epoch gives an overall measure of the client contribution to the FL process for that epoch.

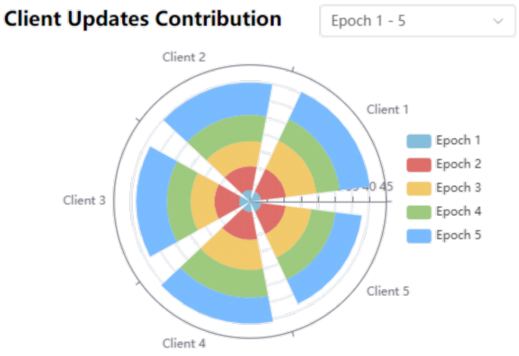

Figure 7: Update parameters polar coordinate chart

b) When an epoch of FL is completed, the parameter update from each client to the server will be visualized. By analyzing the distribution of the updates in the polar graph, users can easily identify patterns and discrepancies among the updates from different clients and can gain insights into the characteristics of the heterogeneous data. It also can help detect Non-IID data among clients.

In the polar graph, the distribution of the updates can be analyzed to determine if there is a significant variation in the clients' contributions to the global model. If clients with similar data distributions produce similar updates, their data can be considered IID. However, if the updates from different clients are widely distributed across the polar graph without clear clustering, it may indicate that the clients have non-IID data.

Moreover, the polar graph can also show if certain clients have significantly different update magnitudes or directions compared to others, which further suggests non-IID data. By detecting non-IID data among clients, users can adjust the aggregation method to ensure a more balanced contribution from each client, which can improve the accuracy and fairness of the resulting global model.

4.7 Model Performance

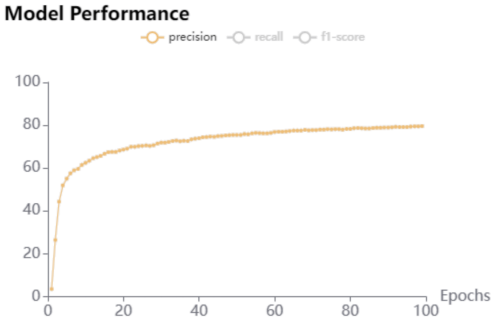

Figure 8: Performance

In the context of NLP-FL tasks, the textual data is inherently

complex, characterized by a wide range of text styles, languages, and vocabularies. This poses significant challenges to the evaluation of the model's performance, as it is difficult to identify the specific factors that contribute to the model's performance. Hence, a comprehensive evaluation system is necessary to analyze and understand the model's performance.

To address this, we have designed a model training performance window that provides researchers with real-time monitoring of the model's performance, including accuracy, recall, and F1 scores. This allows researchers to easily detect potential issues with the model, such as overfitting or underfitting, and to adjust the model parameters timely. Furthermore, this also enables researchers to compare the performance of different models and select the most effective one for their specific NLP-FL task.

Our evaluation metrics offer a comprehensive view of the model's performance. Accuracy, recall, and F1 scores enable researchers to measure the model's ability to correctly predict the target variable, identify the proportion of positive instances that are correctly identified, and balance the trade-off between precision and recall. This comprehensive evaluation system helps researchers to gain a deeper understanding of their model's strengths and weaknesses and optimize the model accordingly.

5 Evaluation

We conducted half-hour interviews with three professionals, namely s1, s2, and s3, who have more than 5 years of experience in the field of federated learning. During the interview, we demonstrated our system and provided a detailed introduction to the experts. After gaining a comprehensive understanding of the system, the experts were free to explore and evaluate it. At the end of each interview, we collected feedback from the experts on three areas: usability, user experience, and advice.

Usability: Regarding usability, the experts were presented with a highly transparent NLP federated learning model and reached a consensus that our system could be instrumental in exploring and designing more efficient NLP models with heterogeneous data. In particular, S2 found the visualization tool to closely simulate a real-world federated learning environment, providing sufficient parameters for visualizing the impact of data and parameter changes on the model in an intuitive manner. Overall, our system was considered user-friendly and effective in enhancing the understanding and optimization of NLP models in a federated learning setting.

User Experience: The user experience received positive feedback from the experts, who appreciated the ability to display the global model parameter update returned by each client in a polar coordinate plot after each training round. S1 and S3 praised the feature that displayed the global model parameter updates returned by each client after each training epoch in a polar coordinate plot. S1 specifically noted its ability to quickly identify clients with significant impact on the model, facilitating adjustments to the aggregation strategy without tweaking the training parameters. In terms of visual appeal, all three experts found the neural network diagram in the center to be eye-catching. Overall, this feature greatly enhanced their ability to monitor and adjust the federated learning process.

Suggestions: While acknowledging the potential usefulness of our system, the experts also offered constructive feedback and recommendations for its improvement. S1 suggested enhancing the interactivity of the interface for observing client parameters by adding a toolbar or other components to the right side of the screen, allowing for changes in visualization settings to dynamically affect the other visualizations on the page. Additionally, S1 recommended using animations and other visual aids to better visualize changes in the global model resulting from client parameter updates. S2 advised expanding the system's audience by incorporating additional NLP models. Taking these suggestions into account could further enhance the usability and effectiveness of our system in future applications.

6 Limitation

In this section, we critically examine the limitations of the current NLP-FL system and outline future research directions based on expert opinions.

Task Limitation: One of the primary limitations of our tool is that it currently only supports the NER task. While NER is a critical NLP component, other NLP tasks such as text classification, sentiment analysis, and machine translation are crucial in many real-world applications. These tasks produce various results, some of which are challenging to evaluate making their integration into our current system difficult. Additionally, our system is currently limited to only four datasets, which may not adequately cover the diversity of NLP tasks. Future research efforts can address this limitation by expanding our platform to support additional NLP tasks and datasets. By doing so, our system could provide a more comprehensive framework for FL research covering a range of NLP tasks.

Scalability Limitation: Our system is designed for scenarios with a limited number of clients and struggles to visualize larger volumes of data. As an increasing number of clients are added, the interface can become cluttered, making it challenging to extract meaningful information. Therefore, creating a scalable framework that accommodates a large number of clients represents a critical area of future research. Such a framework must support many clients while providing an intuitive and user-friendly interface that allows users to access essential information quickly. Future work could explore incorporating features such as panning and zooming to enable users to navigate the interface efficiently or implementing advanced visualization techniques, such as dimensionality reduction or clustering, to improve data processing capabilities. Ultimately, these enhancements would enable our system to address a broader range of FL scenarios, supporting more significant datasets and ensuring high levels of scalability.

7 Conclusion

In the field of NLP, FL has emerged as a promising approach to address the challenges of data privacy and heterogeneity. However, FL models require careful design and development to ensure their effectiveness. Evaluating the impact of textual heterogeneous data on FL models is a particularly challenging task, as it involves analyzing complex data structures and interactions among components of the model.

To address this challenge, we have developed a visualization system that provides researchers with a range of visualizations to aid in exploring the impact of textual heterogeneous data on FL models. These tools enable users to examine parameter updates returned by clients, data distribution, and the state of the neural network during training, thereby providing valuable insights into the behavior of the model and the effects of data and parameter changes.

An expert who has reviewed our visualization tools has validated their effectiveness in facilitating the design and development of more efficient FL models for NLP tasks. Our visualizations enable users to explore the impact of textual heterogeneous data on FL models, empowering them to make informed decisions. By leveraging these visualizations, our approach holds the potential to enhance the development and deployment of FL models in NLP, promising improved performance and efficiency.

References

[1] P. Basu, T. S. Roy, R. Naidu, and Z. Muftuoglu. Privacy enabled financial text classification using differential privacy and federated learning. *arXiv preprint arXiv:2110.01643*, 2021.

[2] D. Cai, Y. Wu, S. Wang, F. X. Lin, and M. Xu. Autofednlp: An efficient fednlp framework. *arXiv preprint arXiv:2205.10162*, 2022.

[3] D. Cai, Y. Wu, H. Yuan, S. Wang, F. X. Lin, and M. Xu. Towards practical few-shot federated nlp. 2023.

[4] Z. Chen, W. Liao, K. Hua, C. Lu, and W. Yu. Towards asynchronous federated learning for heterogeneous edge-powered internet of things. *Digital Communications and Networks*, 7(3):317–326, 2021.

[5] A. B. de Luca, G. Zhang, X. Chen, and Y. Yu. Mitigating data heterogeneity in federated learning with data augmentation. *arXiv preprint arXiv:2206.09979*, 2022.

[6] E. Guberović, C. Alexopoulos, I. Bosnić, and I. Čavrak. Framework for federated learning open models in e-government applications. *Interdisciplinary Description of Complex Systems: INDECS*, 20(2):162–177, 2022.

[7] W. Hao, M. El-Khamy, J. Lee, J. Zhang, K. J. Liang, C. Chen, and L. C. Duke. Towards fair federated learning with zero-shot data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3310–3319, 2021.

[8] S.-M. Lee and J.-L. Wu. Fedua: An uncertainty-aware distillation-based federated learning scheme for image classification. *Information*, 14(4):234, 2023.

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[10] B. Y. Lin, C. He, Z. Zeng, H. Wang, Y. Huang, C. Dupuy, R. Gupta, M. Soltanolkotabi, X. Ren, and S. Avestimehr. Fednlp: Benchmarking federated learning methods for natural language processing tasks. *arXiv preprint arXiv:2104.08815*, 2021.

[11] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, and D. Dou. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, 64(4):885–917, 2022.

[12] M. Liu, S. Ho, M. Wang, L. Gao, Y. Jin, and H. Zhang. Federated learning meets natural language processing: a survey. *arXiv preprint arXiv:2107.12603*, 2021.

[13] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu. Analyzing the training processes of deep generative models. *IEEE transactions on visualization and computer graphics*, 24(1):77–87, 2017.

[14] G. Long, Y. Tan, J. Jiang, and C. Zhang. Federated learning for open banking. In *Federated Learning: Privacy and Incentive*, pp. 240–254. Springer, 2020.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

[16] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, and Z. Zhang. Fedproc: Prototypical contrastive federated learning on non-iid data. *Future Generation Computer Systems*, 143:93–104, 2023.

[17] D. E. O'Leary. Exploiting big data from mobile device sensor-based apps: Challenges and benefits. *MIS Quarterly Executive*, 12(4), 2013.

[18] H. Qin, G. Chen, Y. Tian, and Y. Song. Improving federated learning for aspect-based sentiment analysis via topic memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3942–3954, 2021.

[19] T. Roosta, P. Passban, and A. Chadha. Communication-efficient federated learning for neural machine translation. *arXiv preprint arXiv:2112.06135*, 2021.

[20] D. Sarkar, A. Narang, and S. Rai. Fed-focal loss for imbalanced data classification in federated learning. *arXiv preprint arXiv:2011.06283*, 2020.

[21] A. Z. Tan, H. Yu, L. Cui, and Q. Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[22] J. Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

[23] J. Vig. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 37–42. Association for Computational Linguistics, Florence, Italy, July 2019. doi: 10.18653/v1/P19-3007

[24] J. Vig and Y. Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

[25] J. Wang, L. Gou, H. Yang, and H.-W. Shen. Ganviz: A visual analytics approach to understand the adversarial game. *IEEE transactions on visualization and computer graphics*, 24(6):1905–1917, 2018.

[26] X. Wang, W. Chen, J. Xia, Z. Wen, R. Zhu, and T. Schreck. Hetvis: A visual analysis approach for identifying data heterogeneity in horizontal federated learning. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):310–319, 2022.

[27] X. Wei, Q. Li, Y. Liu, H. Yu, T. Chen, and Q. Yang. Multi-agent visualization for explaining federated learning. In *IJCAI*, pp. 6572–6574, 2019.

[28] Y. Xie, A. Li, L. Gao, and Z. Liu. A heterogeneous ensemble learning model based on data distribution for credit card fraud detection. *Wireless Communications and Mobile Computing*, 2021:1–13, 2021.

[29] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5:1–19, 2021.

[30] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[31] Y. Yeganeh, A. Farshad, N. Navab, and S. Albarqouni. Inverse distance aggregation for federated learning with non-iid data. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2*, pp. 150–159. Springer, 2020.

[32] J. Zhang and Y. Jiang. A data augmentation method for vertical federated learning. *Wireless Communications and Mobile Computing*, 2022:1–16, 2022.

[33] K. Zhang, Z. Cai, D. Seo, et al. Privacy-preserving federated graph neural network learning on non-iid graph data. *Wireless Communications and Mobile Computing*, 2023, 2023.

[34] J. Zhao, J. Wang, Z. Li, W. Yuan, and S. Matwin. Vertically federated learning with correlated differential privacy. *Electronics*, 11(23):3958, 2022.