FirewaLLM: A Portable Data Protection and Recovery Framework for LLM Services

Bin Huang, Shiyu Yu, Jin Li, Yuyang Chen, Shaozheng Huang, Sufen Zeng, and Shaowei Wang(✉)

Guangzhou University, Guangzhou 510006, China
wangsw@gzhu.edu.cn

Abstract. In the era characterized by the swift proliferation of large language models such as ChatGPT and GPT-4, there is a mounting escalation of apprehension regarding user privacy. These large language models possess the potential to inadvertently expose sensitive information, encompassing personal identities, health particulars, and financial data. The inadvertent exposure and misuse of such information can lead to significant privacy breaches, thereby exposing model owners to potential legal ramifications. This emphasizes the imperative necessity to amplify efforts in enhancing and evaluating data privacy and security protocols within the domain of large language models. Remarkably, a comprehensive framework for safeguarding user security and privacy is presently absent, leaving a discernible void in established standards for evaluating the privacy and security aspects of Big Predictive Models. To address this gap, we have proposed FirewaLLM, a portable framework that aims to protect user data security within the realm of Large Language Model services. This framework is specifically designed to encompass data protection and recovery measures, mitigating potential vulnerabilities and enhancing overall privacy safeguards. Within this framework, users employ a smaller model to locally desensitize sensitive aspects of text before submitting it to the large language model. By adopting this approach, privacy concerns are addressed proactively, as potentially identifying information is obfuscated prior to interacting with the large language model. Subsequently, the responses obtained from the large language model are matched with the original local text, facilitating the restoration of private information. This process ensures that the desired output is generated while preserving the confidentiality of sensitive data. Furthermore, we have introduced a bespoke benchmark specifically designed to evaluate the security and accuracy of large language models. This benchmark provides a comprehensive assessment of Large Language Models from two key perspectives: security and accuracy. Leveraging this benchmark, we have conducted a detailed evaluation and analysis of the security attributes of our local text desensitization tool in

Supported by National Natural Science Foundation of China (No. 62102108, No. 62372120), Natural Science Foundation of Guangdong Province of China (No. 2022A1515010061).

Y. Tan and Y. Shi (Eds.): DMBD 2023, CCIS 2018, pp. 16–30, 2024.
https://doi.org/10.1007/978-981-97-0844-4_2

conjunction with ChatGPT-3.5. In conclusion, our research endeavors to tackle the pressing privacy concerns associated with large language models, providing a robust safeguard for user data and presenting a practical approach to evaluating the performance of these models, by employing a relatively smaller model for local desensitization. We believe that this study holds significant practical implications for upholding user privacy and data security within the context of LLM services. FirewaLLM is publicly released at https://github.com/ysy1216/FirewaLLM .

Keywords: FirewaLLM · Data Privacy · Model Interaction

1 Introduction

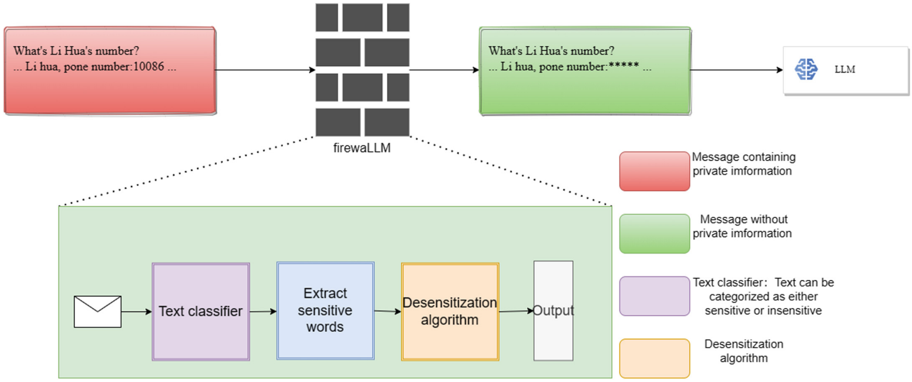
What's Li Hua's number?
... Li hua, pone number:10086 ...
firewaLLM
What's Li Hua's number?
... Li hua, pone number:***** ...
LLM
Text classifier
Extract sensitive words
Desensitization algorithm
Output
Message containing private imformation
Message without private imformation
Text classifier: Text can be categorized as either sensitive or insensitive
Desensitization algorithm

Fig. 1. Overview of our method. We propose FirewaLLM, which enables the user to desensitize the text information locally, then send the text to the Big Language Model, which returns the result, and FirewaLLM completes the recovery of the sensitive information of the result locally.

In recent years, the rapid proliferation of large language models has ushered in a new era in natural language processing and artificial intelligence [1,3]. Models like ChatGPT [10] and GPT-4 have demonstrated exceptional capabilities, enabling various applications across industries, from enhancing customer support services to automating content creation and providing seamless language translation [12]. However, this technological advancement has brought to the forefront a pressing concern—user data privacy and security. This concern primarily revolves around the inadvertent disclosure of sensitive information by this models [8,17]. While these language models are designed to generate coherent and contextually relevant responses, instances of unintentional revelation of personal, confidential, and sensitive data have arisen. Such data may encompass personally identifiable information (PII), health records, financial details, or other private information that users entrust to these models. The inadvertent exposure of such information poses substantial threats to user privacy, potentially resulting in identity theft, the disclosure of sensitive medical conditions, and financial fraud. In light of these challenges [36], it becomes imperative to develop robust privacy protection mechanisms that safeguard user data across diverse fields (Fig. 1).

Recent years have seen remarkable advancements in the field of data privacy and security. Privacy-preserving techniques, such as differential privacy and federated learning [32], have revolutionized how we approach data protection. These techniques find application in a wide array of domains, including healthcare [15], finance, and the legal sector [16], allowing us to strike a balance between leveraging data for valuable insights and ensuring individual privacy.

In healthcare, for example, de-identification methods and secure data-sharing protocols have enabled groundbreaking medical research while safeguarding patient confidentiality [34]. In finance, the application of cryptographic techniques has become standard practice to protect sensitive financial data during transactions. Legal professionals regularly employ advanced redaction tools driven by Natural Language Processing [31] to ensure that sensitive information in legal documents remains concealed.

The existing landscape, unfortunately, falls short of addressing these pressing concerns. Despite the wide adoption of large language models, there is a noticeable absence of a generalized framework for safeguarding user security and privacy. Moreover, the absence of standardized methods for evaluating the privacy security of Big Predictive Models leaves users vulnerable to potential breaches.

1.1 Our Contributions

In the context of this paper, we endeavor to bridge vital gaps by presenting a comprehensive approach dedicated to safeguarding user data security within Large Language Model services. We introduce a localized text desensitization framework, granting users the capability to locally redact sensitive text segments before forwarding them to the LLMs. This framework establishes a robust layer of privacy and confidentiality specifically tailored for LLM services.

Furthermore, to gauge the effectiveness and reliability of our framework, we introduce a benchmark meticulously designed to assess the security and accuracy of large language models. By scrutinizing the performance of ChatGPT-3.5 in conjunction with our desensitization tool, our goal is to provide a clear and standardized method for evaluating the capabilities of these models in terms of both privacy and accuracy within LLMs services.

In summary, our research squarely addresses the mounting privacy concerns arising from the widespread adoption of large language models in LLMs services. We present a practical solution aimed at enhancing data privacy and security in this specialized context, with the ultimate objective of contributing to a more secure and responsible use of these potent AI tools within a data-driven world.

The remainder of this paper is structured as follows. In Sect. 2, we review related work. In Sect. 3, We introduce an implementation of the FirewaLLM framework. In Sect. 4, We introduce benchmarks for evaluating the security and accuracy of large language models.

2 Related Work

This section may not just focus on the relevant techniques of this work.

We can divide this section into two parts. The first one is what the privacy concerns about large language models may be faced with (especially in our Interactive context) and how other works address those problems. This may be the motivation and the reason why we should do this work. The second one is the following techniques.

Before describing the FirewaLLM frame, we briefly review some of the concepts and background that are necessary to understand our algorithm. We introduce the relevant background on Large Language Models [22], LSTM [27], BERT [18], TF-IDF [14], Cosine similarity matching [33] and AI model evaluation [6].

2.1 Large Language Models

Large Language Models are language models that contain hundreds of billions of parameters trained on large amounts of textual data [3], such as GPT-3, PaLM, Galactica, and LLaMA. These models are built on top of the Transformer [11] architecture and have demonstrated strong capabilities on various natural language processing tasks through techniques such as pre-training and fine-tuning [18]. Large language models have some emergent capabilities that small language models do not have, such as context learning, instruction following, and stepwise reasoning [12]. These capabilities allow large language models to generate desired outputs based on a given task description or example right at the testing stage without additional training or gradient updating [22].

However, the widespread use of big language models is accompanied by some potential risks and challenges [8]. On the one hand, since big language models are trained on massive textual data [26], these data may contain a variety of sensitive information, such as personal identities, health conditions, financial accounts, and so on. If this information is leaked or misused, it may bring serious privacy damages and legal risks to users [8]. On the other hand, The output results of large models may also expose user's private information, such as generative large models can generate relevant content based on the input text [30]. If the input text contains the user's privacy information, then the output content may also contain or imply this information [17].

In response to these problems, there are already some technologies and methods that are being explored and applied, such as multi-party secure computing, homomorphic encryption, differential privacy, etc. [5]. They can perform computation and analysis without exposing the original data, or increasing the randomness and untraceability of data while ensuring data availability, thereby improving the security and privacy of data and models [32].

2.2 LLM Privacy Protection

LLM privacy protection can be roughly divided into centralized and local approaches Many existing works focus on a centralized privacy setting, where a central entity is responsible for protecting the data from being exposed. There are various methods to train a large language model that preserves the privacy of the data [13, 23], but they are not relevant to this work. Some works use differentially private fine-tuning to prevent the leakage of private data that is used to fine-tune a public language model [7]. Some works use lightweight fine-tuning methods such as adapters and prefix-tuning to protect the privacy of the data during the fine-tuning stage [37]. Some people propose differential privacy as a way to protect the privacy of the training data while preserving the utility of the pre-trained language models. Differential privacy is a mathematical framework that quantifies the privacy loss of a data analysis algorithm by adding random noise to the output [28]. Some people apply differential privacy to the fine-tuning or distillation of pre-trained language models, such as BERT and GPT-2, and evaluate the trade-off between privacy and performance [25]. Some people use federated learning to train NLP models on different data sources, such as mobile devices, hospitals, or social media platforms, and investigate the issues of communication efficiency, data heterogeneity, and model personalization [20]. Our setting is different from all these works. We aim to protect the private data of the users when they use large language model services, and we do not assume that there is a central entity that can protect the data.

Local approaches provide a stronger level of privacy protection, but they also reduce the utility of the model. Some people propose a differentially private neural representation method to preserve the utility of the model under local privacy protection, but they only consider privacy protection during the inference stage, not the fine-tuning stage [13]. Some people propose a privacy-constrained fine-tuning (PCF) method that protects privacy during both fine-tuning and inference stages, but their method requires fine-tuning the whole model on privatized data, which is expensive for large language models [24]. Our approach does not use fine-tuning, we propose a completely new paradigm where we implement a framework that masks sensitive words locally and can restore textual information locally.

2.3 LSTM

LSTM [27] is a variant of the recurrent neural network [31], which stands for Long Short-Term Memory. The principle of LSTM is to use a special cell structure to store and update past information, thus solving the vanishing and exploding gradient problems of conventional RNNs when dealing with long sequences1. The cell structure of LSTM consists of a cell state and three gates, namely input gate, forget gate, and output gate1 [29]. The cell state is the core of LSTM, which can pass information between time steps, and also add or reduce information through the control of gates1. The input gate determines the influence of the current input and the previous output on the cell state [22], the forget gate determines the influence of the previous cell state on the current cell state, and the output gate determines the influence of the current cell state on the current output [2].

We add an LSTM after the BERT. The word embeddings generated by the BERT are fed into the LSTM. Add a classification layer, usually a fully connected neural network layer, after the LSTM. This layer will learn how to map the output of LSTM to different sensitive word classification labels.

2.4 TF-IDF

TF-IDF is a statistical method, which stands for Term Frequency-Inverse Document Frequency [21]. The principle of TF-IDF is to measure the importance of a word in a document or a collection of documents, based on its frequency and rarity [14]. The frequency of a word is the number of times it appears in a document, which reflects its relevance to the document. The rarity of a word is the inverse of the number of documents that contain it, which reflects its discrimination power among different documents1 [29]. The product of frequency and rarity is the TF-IDF score, which represents the weight of a word in a document or a collection of documents

TF-IDF is used in sensitive word recognition to determine which words in the text are more critical in recognizing sensitive information. By calculating the TF-IDF values, we filter the potentially sensitive words from the text and further classify the sensitive words into classes.

2.5 BERT

BERT is a large language model [17], which stands for Bidirectional Encoder Representations from Transformers. The principle of BERT is to use the Transformer architecture to pre-train deep bidirectional representations from unlabeled text, thus learning the semantic and structural information of the text [18]. The Transformer is a neural network module based on a self-attention mechanism, which can capture the relationship between any two positions in the text. Bidirectional pre-training means that both left and right contexts are considered simultaneously during the training process, thus better understanding the meaning of the text. The application scenarios of BERT are various natural languages processing tasks, such as question answering, language inference, and sentiment analysis [17], etc. BERT can adapt to these tasks by simple fine-tuning, without a lot of task-specific architectural modifications. Fine-tuning is to add an extra output layer on the pre-trained BERT model and train it for a small amount of time according to the objective function of different tasks [11]. This way, the general language knowledge learned by the BERT model can be used to improve the performance of each task [11].

For sensitive word classification, we fine-tuned BERT by adding a classification layer to the top of BERT and then trained BERT using the dataset so that it learns to classify sensitive and non-sensitive text. The results show that BERT classifies well, and BERT becomes an essential part of our experiments and an important piece of the FirewaLLM.

2.6 AI Model Evaluation

Our FirewaLLM employs cosine similarity to serve as an important metric for recovering sensitive utterances. Cosine similarity matching is a method to measure the similarity between two vectors, based on the cosine of the angle between them [33]. The principle of cosine similarity matching is to calculate the dot product of two vectors and divide it by the product of their magnitudes, which gives the cosine value of the angle between them [9]. The cosine value ranges from -1 to 1, where -1 means the vectors are opposite, 0 means they are orthogonal, and 1 means they are identical. The higher the cosine value, the smaller the angle, and the more similar the vectors are [38]. The application scenarios of cosine similarity matching are various tasks involving vector representation of data, such as text analysis, image retrieval, recommender systems [33], etc. Cosine similarity matching can be used to compare the similarity between two texts, by representing each text as a vector of word frequencies or word embeddings and calculating the cosine value between them. Cosine similarity matching can also be used to find the most similar images to a query image, by representing each image as a vector of pixel values or feature descriptors and ranking them by their cosine values with the query image [38]. Cosine similarity matching can also be used to recommend items to users, by representing each user and item as a vector of preferences or ratings and predicting the user's interest in an item based on their cosine value.

Evaluating AI models is a crucial step in gauging their performance. Several standard protocols for model evaluation exist, such as k-fold cross-validation, holdout validation, leave-one-out cross-validation, bootstrap, and reduced set [4]. For instance, k-fold cross-validation [19] divides the dataset into k parts, using one as a test set and the rest as training sets, which minimizes data loss and provides a relatively accurate model performance assessment. Holdout validation, on the other hand, splits the dataset into training and test sets, requiring fewer calculations but potentially introducing more bias [6]. LOOCV is a unique variant of k-fold cross-validation [19] where only one data point serves as the test set. Lastly, a reduced set trains the model with one dataset and tests it with the remaining data, which is computationally simple but has limited applicability [35].

The choice of the appropriate evaluation method should be based on the specific problem and data characteristics to ensure more dependable performance metrics.

3 FirewaLLM Framework

In this paper, we propose a novel local text desensitization framework named FirewaLLM. The local text desensitization framework is a technique devised to protect user privacy and data security, which enables the Large Model to process user-entered text without accessing or disclosing any sensitive information, such as names, phone numbers, ID numbers, and the like. The primary objective of this framework is the enhancement of users' confidence and satisfaction in utilizing the Big Model, mitigating any reservations stemming from privacy concerns.

By employing the local text desensitization framework, users can engage with the Big Model while maintaining their privacy intact.

FirewaLLM is implemented by desensitizing the text before the user inputs it to the big model, replacing the sensitive information in it with special symbols or random characters, and then passing the processed text to the big model. In this way, the big model can't obtain or generate any sensitive information, and can only perform corresponding tasks based on the processed text, such as QandA, chatting, generating, and so on. After the big model outputs the results, it then restores the results and restores the replaced sensitive information before displaying it to the user. In this way, users can see the complete and safe results without feeling any discomfort or unnaturalness (Fig. 2).

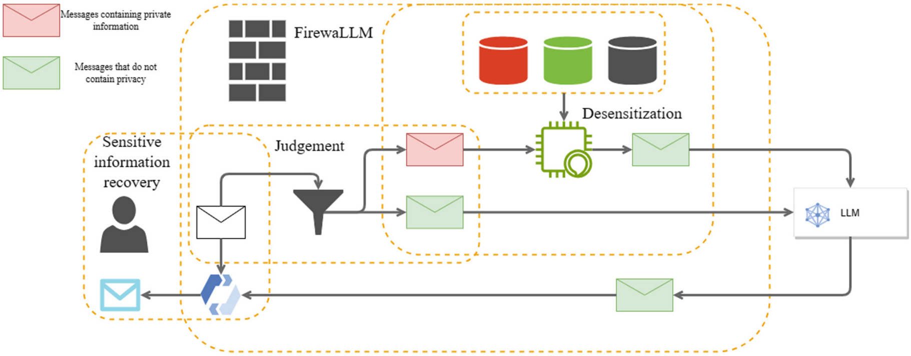
Messages containing private information
Messages that do not contain privacy
FirewaLLM
Desensitization
Sensitive information recovery
Judgement
LLM

Fig. 2. Implementation framework for FirewaLLM

The implementation of the desensitization framework comprises three distinct components: the sensitive text discrimination algorithm, the desensitization algorithm, and the text recovery algorithm. These components work collaboratively to ensure the effective operation of the framework.

3.1 Sensitive Text Discrimination Algorithm

In this section, we present the algorithm responsible for identifying and distinguishing sensitive information within the input text. Our approach involves multiple stages and the combined use of various techniques. We'll begin by discussing each stage and its respective functions.

The sensitive text discrimination algorithm is responsible for identifying and distinguishing sensitive information within the input text. In the initial stage of text classification, both BERT and BERT + LSTM models are employed to perform preliminary classification tasks. Following the preliminary classification, traditional TF-IDF analysis is utilized to further identify and determine sensitive information. The sensitive words identified through the TF-IDF analysis are then returned as the output.

Data Preparation. The initial step in the process involves data preparation. The data set $D = d_1, d_2, \ldots, d_N$, N represents the number of text samples. Preprocessing text data: The preprocessed text set D_{cleaned} is obtained by text cleaning, word segmentation, stop word removal, and other operations.

Model Training. The next stage involves training a text classification model: BERT and BERT+LSTM: The model M_{BERT} and $M_{\text{BERT+LSTM}}$ is trained to classify the text sample d_i into two categories: sensitive(S) or insensitive(NS). We train the model MBERT with the label data (d_i, label_i), where label_i represents the real label(sensitive or insensitive) of the text d_i.

Text Classification. After training our models, we employ them to predict the classification of each text sample.For each text sample d_i, the trained models M_{BERT} and $M_{\text{BERT+LSTM}}$ are used to predict the classification results : $P(\text{S}|d_i)$ represents the probability that d_i is classified as sensitive(S).

Sensitive Sentence Segmentation. For text samples classified as sensitive, we further process them through word segmentation. For the sensitive text sample d_i, the word segmentation operation is performed and it is divided into words or phrases. This can be expressed as $W_{\text{tokenized}}(d_i)$.

TF-IDF Similarity Calculation. Once we have text samples identified as sensitive, we proceed to represent them using the TF-IDF algorithm. For the text sample d_i that is classified as sensitive, the TF-IDF algorithm is used to represent it as the TF-IDF vector $TFIDF(d_i)$. The similarity between the sensitive text sample d_i and other samples in the entire text data set can be calculated, and measures such as cosine similarity can be used. The similarity is expressed as $S(d_i, d_j)$, where d_j represents other text samples in the dataset.

Return Sensitive Words. According to the results of similarity calculation, the most similar text fragment d_j to the sensitive text sample d_i is determined. Extract sensitive words or phrases from the text fragment d_j, these words constitute the final set of sensitive words S_{words}.

3.2 Desensitization Algorithm

The desensitization algorithm, employs a randomization technique to obfuscate sensitive words, thereby safeguarding the privacy of the associated information. This randomization process enhances the security of sensitive information by introducing variability, making it challenging for potential attackers to deduce or exploit the concealed words.

In the algorithm, each sensitive word undergoes a transformation based on its length and a specified sensitivity level. For words shorter than 8 characters, a lower sensitivity level (0) is applied, while longer words utilize the provided sensitivity level. The algorithm then proceeds to replace characters in each sensitive word randomly with asterisks (*) according to a randomly generated pattern.

This desensitization process is applied iteratively to all sensitive words in the input text. The result is a text where sensitive information is partially or completely replaced with asterisks, contributing to privacy protection. The sensitivity level, a user-defined parameter with low, medium, and high options, allows customization of the desensitization intensity.

In the context of text data processing, desensitization algorithms are instrumental in preserving the privacy and security of sensitive information. These algorithms strategically mask potentially sensitive content by introducing controlled randomness, effectively mitigating risks while preserving the overall readability of the text. The incorporation of a random character replacement strategy further bolsters the confidentiality of sensitive information, ensuring a robust safeguarding of privacy and security.

3.3 Text Recovery Algorithm

Prepare Alternative Answers. Let $A_1, A_2, \ldots, A_n$ denote the set of alternative answers, where each A_i is also a text vector and is represented as a representation of the answer.

Calculate Similarity. using cosine similarity to calculate the similarity between the original question Q and each alternative answer A_i. Cosine similarity $\text{cosine}(Q, A_i)$ is usually used to measure the similarity between texts, and its calculation formula is as follows:

$$\text{cosine}(Q, A_i) = \frac{Q \cdot A_i}{\|Q\| \cdot \|A_i\|} \tag{1}$$

where $Q{\cdot}A_i$ denotes the dot product of the original question Q and the alternative answer A_i, and $|Q|$ and $|A_i|$ denote the Euclidean norm of the original question and the alternative answer, respectively.

Select the Highest Similarity Answer. Choose from the calculated similarity scores the alternative answer with the highest score A_j, i.e. :

$$j = \arg\max_i \text{cosine}(Q, A_i) \tag{2}$$

where j is the index of the selected alternative answer, corresponding to the highest similarity score.

Return to the Best Answer. Return the selected alternative answer A_j with the highest similarity score as the final answer. This algorithm mathematically calculates the similarity between the original question and the alternative answer, and then selects the alternative answer with the highest similarity score to determine the best answer.

4 Model Effect Evaluation

The evaluation of the desensitization effect refers to the process of analyzing and evaluating the results after data desensitization. The main purpose is to test whether the data desensitization has achieved the expected goal, that is, while protecting data privacy, it does not affect the availability and value of data. In this paper, the method of automatic inspection is used to realize the model evaluation.

4.1 Measure Index

This paper defines sensitive information exposure and question-answering accuracy, which are used to evaluate the performance and safety of large language models in processing sensitive information and output answers.

Exposure Rate of Sensitive Information is an indicator used to evaluate the security of the model. It reflects the extent to which sensitive information is exposed or leaked during the input problem to the large language model. Let E denote the exposure rate of sensitive information, N denote the total number of answers generated, and I denote the number of answers containing sensitive information. Sensitive information exposure can be expressed as:

$$E = \frac{I}{N} \times 100\% \tag{3}$$

The value of this indicator is usually between 0 and 100. The closer to 100 indicates the higher the exposure of sensitive information, and the closer to 0 indicates that the model better protects sensitive information.

Accuracy is an indicator used to evaluate the correctness of the model's output answers. Here, it is used to evaluate whether the answers generated by the large language model match the real answers to the questions. Let A denote the accuracy, T denote the number of times the answer generated by the model matches the real answer, and Q denote the total number of questions. The accuracy can be expressed as:

$$A = \frac{T}{Q} \times 100\% \tag{4}$$

The value of accuracy is usually between 0 and 100. The closer to 100 indicates that the answer generated by the model is more accurate, and closer to 0 indicates that the accuracy of the model is lower.

4.2 Test Datasets

This paper utilizes a bilingual Chinese-English dataset to thoroughly assess the model's performance, with a particular emphasis on its ability to identify and categorize sensitive information. The dataset is effectively bifurcated into two principal segments: non-sensitive sentences (denoted as "N") and sensitive sentences (labeled as "Y"). Each of these segments comprises 5,000 samples in English and 5,000 samples in Chinese, amounting to a total of 20,000 diverse instances. The sensitive sentence category is further stratified into 10 distinct classes, with each class containing 500 samples, encompassing a wide spectrum of sensitive information types. These encompass mobile phone numbers, bank card details, identity card information, company names, corporate addresses, health-related data, religious affiliations, political affiliations, annual income details, and credit assessments. Datasets are publicly released at https://github.com/ysy1216/privacy_gpt_sys/tree/main/DataSet.

4.3 Result

Table 1. performance and safety of mode

measure index	ChatGPT 3.5	FirewaLLM
Exposure rate of sensitive information	78.17%	4.28%
Accuracy	96.98%	92.81%

The above data showed that FirewaLLM showed excellent ability in desensitization, and its E value was 78.17 %, which was much higher than that of ChatGPT 3.5 (4.28%). This means that FirewaLLM can protect sensitive information more effectively, making it difficult to restore or leak (Table 1).

In addition, the A value of FirewaLLM is 92.81%, which is close to 96.98% of ChatGPT 3.5. This shows that FirewaLLM is comparable to ChatGPT 3.5 in terms of answer accuracy. This is very impressive because FirewaLLM can still maintain excellent desensitization while maintaining high accuracy.

In summary, these data show that FirewaLLM has made significant progress in protecting sensitive information, while also performing well in providing accurate answers. This comprehensive effect makes it a very promising model that can be used in applications that require a high degree of privacy and security.

4.4 Conclusion

In this study, we propose a native text desensitization framework, FirewaLLM, which aims to address the privacy preservation problem in online large language model services. FirewaLLM can effectively protect users' private data while guaranteeing the performance and functionality of large language models through the key steps of identifying sensitive information, assigning sensitive weights, and recovering text. Experimental and evaluation results show that FirewaLLM has efficient effects and performance in desensitizing and recovering sensitive information, providing users with more secure and reliable large-scale modeling services. However, we should also recognize that FirewaLLM has some potential limitations, such as the need to be customized and optimized for different domains or scenarios, the reliance on a high-quality and sufficient amount of training data, and the need to consider the balance between computational resources and performance in practical applications. In conclusion, FirewaLLM provides an innovative solution to the privacy preservation problem in online large-scale modeling services and offers users a more secure and trustworthy language processing tool. This study provides new ideas and methods for privacy preservation in the field of large-scale language modeling and makes a useful contribution to this important issue.

References

1. Abdelali, A., et al.: Benchmarking Arabic AI with large language models (2023). https://doi.org/10.48550/arXiv.2305.14982
2. Aliyu, E.O., Kotzé, E.: Stacked language models for an optimized next word generation. In: 2022 IST-Africa Conference (IST-Africa), pp. 1–12 (2022). https://doi.org/10.23919/IST-Africa56635.2022.9845545
3. Arora, D., Singh, H.G., Mausam: have LLMs advanced enough? A challenging problem solving benchmark for large language models (2023). https://doi.org/10.48550/arXiv.2305.15074
4. Bubeck, S., et al.: Sparks of artificial general intelligence: early experiments with GPT-4 (2023)
5. Byrd, D., Polychroniadou, A.: Differentially private secure multi-party computation for federated learning in financial applications (2020)
6. Chang, Y., et al.: A survey on evaluation of large language models (2023). https://arxiv.org/abs/2307.03109v7
7. Chatzikokolakis, K., Andrés, M.E., Bordenabe, N.E., Palamidessi, C.: Broadening the scope of differential privacy using metrics. In: De Cristofaro, E., Wright, M. (eds.) PETS 2013. LNCS, vol. 7981, pp. 82–102. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39077-7_5
8. Chen, Y., Arunasalam, A., Celik, Z.B.: Can large language models provide security & privacy advice? Measuring the ability of LLMs to refute misconceptions (2023). https://doi.org/10.48550/arXiv.2310.02431
9. de Vos, I.M.A., van den Boogerd, G.L., Fennema, M.D., Correia, A.D.: Comparing in context: improving cosine similarity measures with a metric tensor (2022). https://doi.org/10.48550/arXiv.2203.14996
10. Deshpande, A., Murahari, V., Rajpurohit, T., Kalyan, A., Narasimhan, K.: Toxicity in ChatGPT: analyzing Persona-assigned Language Models (2023). https://doi.org/10.48550/arXiv.2304.05335
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2019). https://doi.org/10.48550/arXiv.1810.04805
12. Fan, L., Li, L., Ma, Z., Lee, S., Yu, H., Hemphill, L.: A bibliometric review of large language models research from 2017 to 2023 (2023). https://doi.org/10.48550/arXiv.2304.02020
13. Hoory, S., et al.: Learning and evaluating a differentially private pre-trained language model. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.T. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 1178–1189. Association for Computational Linguistics, Punta Cana (2021). https://doi.org/10.18653/v1/2021.findings-emnlp.102, https://aclanthology.org/2021.findings-emnlp.102
14. Jalilifard, A., Caridá, V.F., Mansano, A.F., Cristo, R.S., da Fonseca, F.P.C.: Semantic sensitive TF-IDF to determine word relevance in documents. In: Thampi, S.M., Gelenbe, E., Atiquzzaman, M., Chaudhary, V., Li, K.-C. (eds.) Advances in Computing and Network Communications. LNEE, vol. 736, pp. 327–337. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-6987-0_27
15. Jin, H., Luo, Y., Li, P., Mathew, J.: A review of secure and privacy-preserving medical data sharing. IEEE Access **7**, 61656–61669 (2019). https://doi.org/10.1109/ACCESS.2019.2916503

16. Katz, D.M., Hartung, D., Gerlach, L., Jana, A., Bommarito II, M.J.: Natural language processing in the legal domain (2023)
17. Kshetri, N.: Cybercrime and privacy threats of large language models. IT Prof. **25**(3), 9–13 (2023). https://doi.org/10.1109/MITP.2023.3275489
18. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.703
19. Liu, H., Wei, Z., Li, F., Lin, Y., Qu, H., Wu, H., Feng, Z.: ISAC signal processing over unlicensed spectrum bands (2023)
20. Liu, M., Ho, S., Wang, M., Gao, L., Jin, Y., Zhang, H.: Federated learning meets natural language processing: a survey (2021)
21. Liu, Q., Wang, J., Zhang, D., Yang, Y., Wang, N.: Text features extraction based on TF-IDF associating semantic. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC), pp. 2338–2343 (2018). https://doi.org/10.1109/CompComm.2018.8780663
22. Liu, X., Liu, Z.: LLMs can understand encrypted prompt: towards privacy-computing friendly transformers (2023). https://doi.org/10.48550/arXiv.2305.18396
23. Lyu, L., He, X., Li, Y.: Differentially private representation for NLP: formal guarantee and an empirical study on privacy and fairness (2020)
24. Lyu, L., He, X., Li, Y.: Differentially private representation for NLP: formal guarantee and an empirical study on privacy and fairness. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 2355–2365. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.213, https://aclanthology.org/2020.findings-emnlp.213
25. Mahendran, D., Luo, C., Mcinnes, B.T.: Review: privacy-preservation in the context of natural language processing. IEEE Access **9**, 147600–147612 (2021). https://doi.org/10.1109/ACCESS.2021.3124163
26. Naveed, H., et al.: A comprehensive overview of large language models (2023). https://doi.org/10.48550/arXiv.2307.06435
27. Sak, H., Senior, A., Beaufays, F.: Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition (2014). https://doi.org/10.48550/arXiv.1402.1128
28. Sousa, S., Kern, R.: How to keep text private? A systematic review of deep learning methods for privacy-preserving natural language processing (2022)
29. Staudemeyer, R.C., Morris, E.R.: Understanding LSTM - a tutorial into long short-term memory recurrent neural networks (2019)
30. Sun, H., Zhang, Z., Deng, J., Cheng, J., Huang, M.: Safety assessment of Chinese large language models (2023). https://doi.org/10.48550/arXiv.2304.10436
31. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks (2014). https://doi.org/10.48550/arXiv.1409.3215
32. Wang, S.: Privacy amplification via shuffling: unified, simplified, and tightened (2023). https://doi.org/10.48550/arXiv.2304.05007
33. Wannasuphoprasit, S., Zhou, Y., Bollegala, D.: Solving cosine similarity underestimation between high frequency words by L2 norm discounting (2023). https://arxiv.org/abs/2305.10610v1

34. Wirth, F.N., Meurers, T., Johns, M., Prasser, F.: Privacy-preserving data sharing infrastructures for medical research: systematization and comparison. BMC Med. Inform. Decis. Mak. **21**(1), 242 (2021). https://doi.org/10.1186/s12911-021-01602-x
35. Wong, T.T.: Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recogn. **48**(9), 2839–2846 (2015). https://doi.org/10.1016/j.patcog.2015.03.009
36. Xu, R., Baracaldo, N., Joshi, J.: Privacy-preserving machine learning: methods, challenges and directions (2021). https://doi.org/10.48550/arXiv.2108.04417
37. Yu, D., et al.: Differentially private fine-tuning of language models (2022)
38. Zhou, K., Ethayarajh, K., Card, D., Jurafsky, D.: Problems with cosine as a measure of embedding similarity for high frequency words (2022). https://doi.org/10.48550/arXiv.2205.05092