Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 1727285, 23 pages
https://doi.org/10.1155/2023/1727285

Research Article

An Automatic Privacy-Aware Framework for Text Data in Online Social Network Based on a Multi-Deep Learning Model

Gan Liu,[1] **Xiongtao Sun**,[2] **Yiran Li**,[2] **Hui Li**,[2] **Shuchang Zhao**,[2] **and Zhen Guo**[1]

[1]*School of Cyberspace Security (School of Cryptology), Hainan University, Haikou, China*
[2]*School of Cyber Engineering, Xidian University, Xi'an, China*

Correspondence should be addressed to Hui Li; lihui@mail.xidian.edu.cn

Received 5 December 2022; Revised 31 July 2023; Accepted 9 October 2023; Published 8 November 2023

Academic Editor: Yu-an Tan

With the increasing severity of user privacy leaks in online social networks (OSNs), existing privacy protection technologies have difficulty meeting the diverse privacy protection needs of users. Therefore, privacy-aware (PA) for the text data that users post on OSNs has become a current research focus. However, most existing PA algorithms for OSN users only provide the types of privacy disclosures rather than the specific locations of disclosures. Furthermore, although named entity recognition (NER) technology can extract specific locations of privacy text, it has poor recognition performance for nested and interest privacy. To address these issues, this paper proposes a PA framework based on the extraction of OSN privacy information content. The framework can automatically perceive the privacy information shared by users in OSNs and accurately locate which parts of the text are leaking sensitive information. Firstly, we combine the roformerBERT model, BI_LSTM model, and global_pointer algorithm to construct a direct privacy entity recognition (DPER) model for solving the specific privacy location recognition and entity nesting problems. Secondly, we use the roformerBERT model and UniLM framework to construct an interest privacy inference (IPI) model for interest recognition and to generate interpretable text that supports this interest. Finally, we constructed a dataset of 13,000 privacy-containing texts for experimentation. Experimental results show that the overall accuracy of the DPER model can reach 91.80%, while that of the IPI model can reach 98.3%. Simultaneously, we compare the proposed model with recent methods. The analysis of the results indicates that the proposed model exhibits better performance than previous methods.

1. Introduction

With the rapid development of Internet technology, a large number of social networking platforms have been established, satisfying the social needs of Internet users. A large number of Internet users have signed up on several social networking platforms through which they can share a multitude of information. In particular, most of these platforms (e.g., Facebook, Twitter, and microblogs) permit users to share their opinions, feelings, snippets of their lives, and political commentaries. At present, social networks play an important role in the daily lives of many people. In this regard, these online networks have changed the way individuals perceive the world, as people now have the convenience of communicating information directly without boundaries [1].

As of May 2022, the total number of online social networking (OSN) users has reached 5.4 billion on more than 300 OSN platforms [2]. For example, Weibo, WeChat, Twitter, Facebook, and other social networking platforms have more than 1 billion users. Most online social platforms encourage users to express themselves through their platforms because users share content that is more attractive to others than professional content, increasing engagement on the platform [3]. Users frequently share information publicly on social networks or public online platforms, which commonly contains a large amount of personal information. However, the indiscriminate spread of such content online can endanger privacy information, consequently exposing users to many risks [4, 5]. For example, sharing travel information may allow burglars to know that you are not at home, giving them the opportunity to break in and steal your belongings. In addition, sharing information about a new house may attract a lot of calls from decoration companies or intermediary telephone promotions. Moreover, sharing labor remuneration may invite telephone fraud. The disclosure of such information causes endless security incidents that range from discrimination or cyberbullying to fraud and identity theft, which affect, and even threaten, the lives of Internet users [6]. Therefore, we must analyze how users manage their privacy needs on social networks to identify which information that involves privacy leakage is of great significance in making users more aware of how to prevent privacy issues.

The leakage of privacy information on social networks has triggered considerable research. The direct approach is to protect privacy information in social networks [7]. Researchers have proposed k-anonymous-based privacy data protection technology [8, 9], data perturbation technology [10, 11], cryptography-based privacy protection technology [12], and differential privacy-based privacy data protection methods [13, 14]. The premise of these technologies is to identify privacy data in social networks and then conduct the corresponding privacy processing. The major technology is to use privacy information scale to identify privacy information. However, this method can only identify specific categories of privacy information because indirect privacy leakage does not exhibit a good protection effect. Simultaneously, some researchers have started from the dynamic characteristics of social networks to protect privacy in social networks [15]. A number of experts have proposed to analyze privacy in dynamic social networks by using privacy propagation and accumulation [16, 17], along with centralized [18] and decentralized technologies [19] for privacy protection. Other researchers have also proposed the use of compressed sensing technology to protect the privacy of dynamic social networks [20]. Although these methods can protect privacy, they cannot meet the specific needs of individuals.

Studies have utilized natural language processing (NLP) technology to censor the content published by users automatically. However, the audit primarily focuses on the automatic censorship of political tendencies, dirty language and hate speech [21–23], false news testing [24], and spam review [25]. These techniques do not involve reviewing personal and sensitive information. Notably, the content shared by network users can be crawled by third-party crawler software and analyzed to obtain the corresponding commercial value. To avoid shared content from being crawled, network users either set permission for the information to be visible to friends only or make the information visible for only 3 days. Although these strategies prevent the spread of personal shared content on the network in time and space, they do not process the sensitive information in the user's text content. Moreover, they can still cause the leakage of the user's sensitive information within a small range. Some Internet users also use automatic disinfection technology to process the shared content. Nevertheless, existing automatic disinfection technologies replace sensitive terms in specific areas (medical neighborhoods and criminal records) with ordinary personal sensitive terms and delete some sensitive terms. These methods exhibit a high degree of ambiguity, and the deleted information may cause poor readability of the original information [26, 27]. The use of deep learning (DL) and machine learning (ML) techniques for privacy classification or the recognition of privacy entities in the text shared by users in social networks has attracted the attention of researchers. Accordingly, some models have been proposed. However, given the complexity of social network text, variation of text length, existence of nested privacy entities, and other problems, these proposed models cannot solve aforementioned issues.

Researchers have studied self-disclosure in social networks, which is mostly done unconsciously [28, 29]. Existing research on self-disclosure behavior primarily utilizes the questionnaire survey for privacy information in specific fields. Then, the risk of the self-disclosure information of users in the corresponding field is obtained by analyzing the questionnaire survey. This method is laborious, and the results obtained cannot be widely used. These self-disclosure studies only provide information on whether a privacy leak occurs. Meanwhile, some studies have classified self-disclosure privacy information into several relatively broad categories, utilizing ML algorithms to predict the categories of user privacy self-disclosure [30]. Other researchers label text as sensitive or nonsensitive and then use the DL model for text classification [31]. These methods only provide qualitative knowledge of user self-disclosure but do not point to specific sensitive information locations of user self-disclosure.

Researchers have also used reasoning attacks to infer privacy about OSNs. Graph perturbation defense graph neural network has been used for privacy reasoning [32]. Bayesian inference and individual privacy difference rules have been adopted to deduce user privacy [33]. Adversarial training techniques, such as overlapping technology, have been applied to deduce and protect the sensitive information of users [34]. These reasoning techniques can only detect specific types of privacy information, not multiple types of privacy leakage reasoning.

To address the aforementioned issues, Li et al. proposed a theoretical framework for privacy computing from the perspective of the entire lifetime of privacy information [35]. The work in the current research belongs to the privacy-aware link of privacy computing, which is the primary component of the whole theoretical framework. Our objectives are as follows: to be able to automatically perceive the text-sensitive information shared in the social network of users, to accurately locate which part of the text is leaking sensitive information, and to send these privacy data as feedback to users to improve their privacy-aware (PA). We propose a framework for automatically identifying privacy entity in social text, as shown in Figure 1. This framework is composed of two parts.

The first part is the direct privacy module, which primarily uses the named entity recognition (NER) method to extract direct privacy entity. Direct privacy entity refers to privacy information that is directly exposed in the text, including basic personal information (e.g., height, weight, birthday, and gender), address (e.g., company address, home address, and current location), job, educational background, and employer/company. In this module, we combine the roformerBERT model, BI_LSTM model, and global_pointer (GP) algorithm to build a direct private entity recognition (DPER) model. This model can not only extract privacy information in the text but also provide the specific location of privacy information. Meanwhile, the recognition ability of nested private entities is enhanced by this model. The RoformerBERT model, based on rotating encoding and proposed by SU, is a variant of the BERT (bidirectional encoder representation from transformers) model. Its primary objective is to enhance the traditional BERT model's capacity to process longer character sequences [36].

The second part is the indirect privacy module. In our experiment, some indirect privacy leaks are difficult to uncover, such as "I want to travel with Anlics, and I will not come back next month." With the NER model, this sentence identifies no sensitive privacy information. However, this sentence discloses the individual's personal interests in travel. The model designed in this study is primarily used to identify privacy information about interests because most the information leakage on interests occurs when users inadvertently share information that can be obtained by attackers. The interest information in this study mostly includes lifestyle, design aesthetics, games, sports, variety shows, film and television, finance and economics, tourism, motherhood, animation, reading, and food. This part combines the roformerBERT model and the UniLM framework to build a user interest privacy inference (IPI) model. The IPI model not only infers which privacy information is leaked in social text but also offers information on which corresponding text causes indirect privacy leakage.

The contributions of this study are summarized as follows:

(1) We propose a PA framework for OSNs that can automatically sense sensitive text information shared by users in social networks and accurately locate which part of the text leaks sensitive information.

(2) We construct two models. To address the problems of nested privacy entities and the unbalanced distribution of privacy entities in social networks, we build a DPER model by combining the roformerBERT model, BI_LSTM model, and GP algorithm. To solve the poor interpretability problem of the existing IPI, a user IPI model is constructed by integrating the roformerBERT model into the UniLM framework.

(3) We construct a new annotation corpus with about 13,000 text data and annotate the privacy information content of each text datum.

The remainder of this paper is organized as follows. Section 2 introduces the related works. Section 3 presents the PA framework composition and the key components of the model. Section 4 describes the details of the experimental design and discusses the experimental results. Section 5 concludes the study.

2. Related Work

This section introduces relevant research from two aspects: the traditional study of personal privacy in social networks and the perception of personal privacy based on NLP and DL.

2.1. Traditional Research on Personal Privacy in Social Networks. Personal privacy information in social networks has long been widely examined by many scholars. Researchers have proposed a variety of different methods that can be generalized into two major research directions.

The first direction is privacy information measurement in social networks [37, 38], while the second one is privacy protection in social networks [39]. In social network privacy information measurement, Buchanan et al. used a questionnaire to compute the privacy scale of multiple dimensions; a reliable and effective social network privacy measurement was eventually obtained by verifying the validity of differential data [40]. Srivastava and Geethakumari surveyed the possible privacy leakage problem in the network world, computed the privacy coefficient of users, and proposed an unstructured privacy measurement model to measure the degree of privacy information leakage in the text data published by users [41]. These privacy measures are comparatively simple and biased toward specific research, and these are difficult to adapt to the current complex network environment. Serfontein et al. utilized a self-organizing map to recognize possible risk in networks [42]. Alsarkal et al. quantified the degree of privacy disclosure that might lead to co-disclosure among friends. By researching the differences between self-disclosure and co-disclosure on various privacy disclosures, users can utilize different protection strategies for various privacy sources [43]. Shi et al. used static network structure entropy in a complex network structure to measure privacy. Defined as the privacy measurement indexes (PMI), it measures the privacy protection ability of a graph structure. Finally, they used PMI to design a graph of a privacy protection classification scheme [44]. This scheme considers users' friends and privacy leakage measures [8]. Nevertheless, if a user has friends, then analyzing each user is inefficient and affects the final privacy measures. At the same time, these measurements simply quantify privacy data in social networks, allowing users to know how much privacy they are leaking. However, users do not know which data have been leaked by their friends, and they cannot take the initiative to protect privacy leakage.

In the case of research on the protection of social information privacy, the researchers proposed a technology for privacy data protection based on k-anonymity, data perturbation technology, cryptography-based privacy protection technology, and differential privacy-based data protection methods. Privacy data protection technology based on k-anonymity mostly applies k-anonymous model to social networks to generalize and hide privacy data [10]. However, this technique does not satisfy the diversity of privacy properties. Data perturbation technology is largely based on the idea of data randomization evolution, which uses data randomization to encrypt sensitive information [12, 45, 46]. Privacy protection technology based on cryptography provides differential social network privacy protection [13, 47, 48] and homomorphic encryption network data privacy protection [49]. The premise of these technologies is to recognize the privacy of data in social networks and then conduct analogous privacy processing [50]. Most of these technologies are used to identify privacy information through a privacy information measurement scale. However, this method can only recognize specific categories of privacy information and cannot exert a good protection effect on indirect privacy leakage.

2.2. Perception of Personal Privacy Based on NLP and DL. With the rise of NLP technology and DL, many scholars have utilized these technologies to analyze privacy data. Various models and methods have been proposed.

Vasalou et al. proposed the concept of a privacy dictionary and designed this dictionary by utilizing NLP technology, traditional privacy theory methods, and prototype theory. Their objective was to help researchers with the automated content analysis of texts, which is a valuable addition to the tools available for privacy research [51]. Gill et al. modified the privacy dictionary proposed by Alastair and used corpus linguistics to construct and validate eight dictionary categories from empirical materials within a wide range of privacy-sensitive contexts. The generated dictionary combined with LIWC software can quickly recognize privacy information in text. Although this privacy dictionary approach can provide high precision, it has poor recall because it relies only on the count of sensitive words in a document, regardless of the context in which the words are used [52].

Xu et al. constructed a text-sensitive content detection model by utilizing Text-CNN in convolutional neural networks (CNNs). Compared with recurrent neural networks, Text-CNN can simultaneously process multiple filters in parallel while ensuring the same detection effect. In addition, the training time of the model is lower, and detection speed is faster when using Text-CNN [53]. Mehdy et al. used NLP to process text and obtain text features, such as linguistic labels, syntactic dependencies, entity relations, and other features. Then, a CNN model was trained with the obtained text features. The trained model can recognize whether text has a privacy leakage risk. Their proposed method is essentially a binary classification model that can recognize whether text has privacy leakage [54]. These methods use the corresponding technology to obtain text features to train the corresponding prediction model. Then, they eventually apply the trained model to the privacy perception of the text. Nevertheless, these methods do not adequately consider the context characteristics of text data and exhibit poor interpretability in prediction. Users only know that privacy leakage exists when using them. However, they do not know which privacy leakage occurs.

Li et al. employed the NER model (BI_LSTM-CRF) to identify privacy entities in Twitter. They divided privacy into four parts, and F1 finally reached 84% [55]. Wu et al. used the DL and ontology models to identify privacy information in Twitter. They also classified four privacy entities and used the privacy ontology model to subdivide the privacy. However, prediction accuracy was not sufficiently accurate, and the recognition accuracy values of event and trait were only 64% and 76%, respectively [56]. Li et al. utilized graph convolutional network (GCN) to measure the privacy leakage of microblog data users. Their method can effectively extract the privacy measure of users in social networks [57]. These research strategies can recognize privacy data. However, the semantics of social networks is complex, and the existence of privacy data must be combined with specific entities before privacy leakage occurs.

3. Methodology

This section presents the design of the privacy-aware (PA) framework for social networks and the key components of both the DPER and IPI models. Before introducing the model, we summarize the main notations in Table 1 to understand the following model calculation process.

3.1. Privacy-Aware (PA) Framework Architecture. To better solve the problem of OSN privacy perception, we designed a new PA framework. This framework consists of the DPER and IPI models. Specifically, this framework is composed of the GP algorithm, BI_LSTM model, roformerBERT model, and UniLM framework. The overall flowchart is presented in Figure 1. The feature representation of the PA framework is provided in formula (1), and we define X_{in} and H_{PA} as the input and output features, respectively, of the PA framework.

$$H_{\text{PA}} = \left[g_{\text{GP}}\left(g_{\text{BL}}\left(g_{\text{RFB}}\left(X_{\text{in}}\right)\right)\right)\colon g_D\left(g_E\left(g_U\left(g_{\text{RFB}}\left(X_{\text{in}}\right)\right)\right)\right)\right], \tag{1}$$

where X_{in} is a feature processed with embedding and [:] represents a connection operation. g_{RFB} represents the roformerBERT model operation that contains rotational position encoding operations. g_{BL} represents the BI_LSTM model operation, which is primarily used to extract the sequence feature information of sentences. g_{GP} represents the GP algorithm operation. g_U represents the actions processed by the UniLM framework. g_D and g_E represent encoding and decoding operations, respectively. Moreover, the activation function used in each DL is the RELU function. The final output layer of each model is processed using the softmax function.

3.2. DPER Model. The DPER model is composed of the BERT pretrained model, the BI_LSTM model, and the GP algorithm. The DPER model feature is represented as shown in formula (2), where H_{DPER} represents the output features

Table 1: The main symbols in the model calculation process.

Symbols	Description
H	Representation of the output of a framework or model, with detailed description provided in the article
X_{in}	The input feature of a sequence
g	The operation of a model or module
$\tilde{q}, \tilde{k}$	Represents the result of adding absolute location information to q and k
f_{Rope}	Represents the RoPE rotary encoding operation
f_{QDense}	Computes the fully connected operation of the query matrix
f_{KDense}	Computes the fully connected operation of the key matrix
w, b	Weights; biases
Q, K, V	Query matrix, key matrix, value matrix
$M_{i,j}$	Mask matrix
A_l	Self-attention head output
P, N	The set of privacy class, the set of nonentity class
$P(.)$	Probability calculation

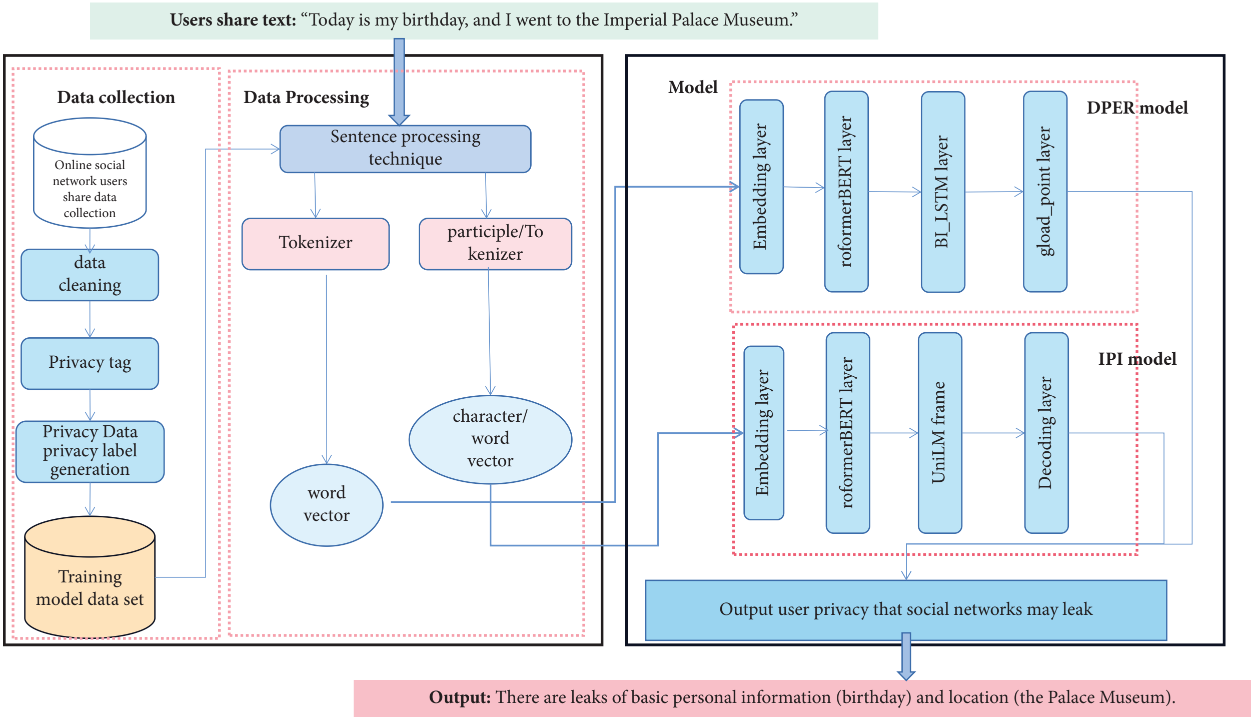
Users share text: "Today is my birthday, and I went to the Imperial Palace Museum."
Data collection
Online social network users share data collection
data cleaning
Privacy tag
Privacy Data privacy label generation
Training model data set
Data Processing
Sentence processing technique
Tokenizer
participle/To kenizer
character/ word vector
word vector
Model
Embedding layer
roformerBERT layer
BI_LSTM layer
gload_point layer
DPER model
Embedding layer
roformerBERT layer
UniLM frame
Decoding layer
IPI model
Output user privacy that social networks may leak
Output: There are leaks of basic personal information (birthday) and location (the Palace Museum).

Figure 1: Overview of the automatic PA framework for text data in OSNs. First, the framework uses NLP technology to transform the features of the obtained text data, including word segmentation and tokenizer operation. Then, we use DL models for privacy entity sensing and inference. We construct the DPER model for privacy entity sensing and the IPI model for IPI. Finally, we combine the calculated values of the two models and send them as feedback to the user.

of the DPER model. g_{RFB} represents the roformerBERT model operation. g_{GP} represents the GP algorithm operation. g_{BL} represents the BI_LSTM model operation.

$$H_{\text{DPER}} = g_{\text{GP}}\left(g_{\text{BL}}\left(g_{\text{RFB}}\left(X_{\text{in}}\right)\right)\right). \tag{2}$$

In particular, the roformerBERT pretraining model is used to train the model, which can not only learn text features deeply but also better solve the imbalance problem of privacy entity distribution. The BI_LSTM model is primarily used to extract the sequence features of sentences. The GP algorithm is used to solve the nested problem of

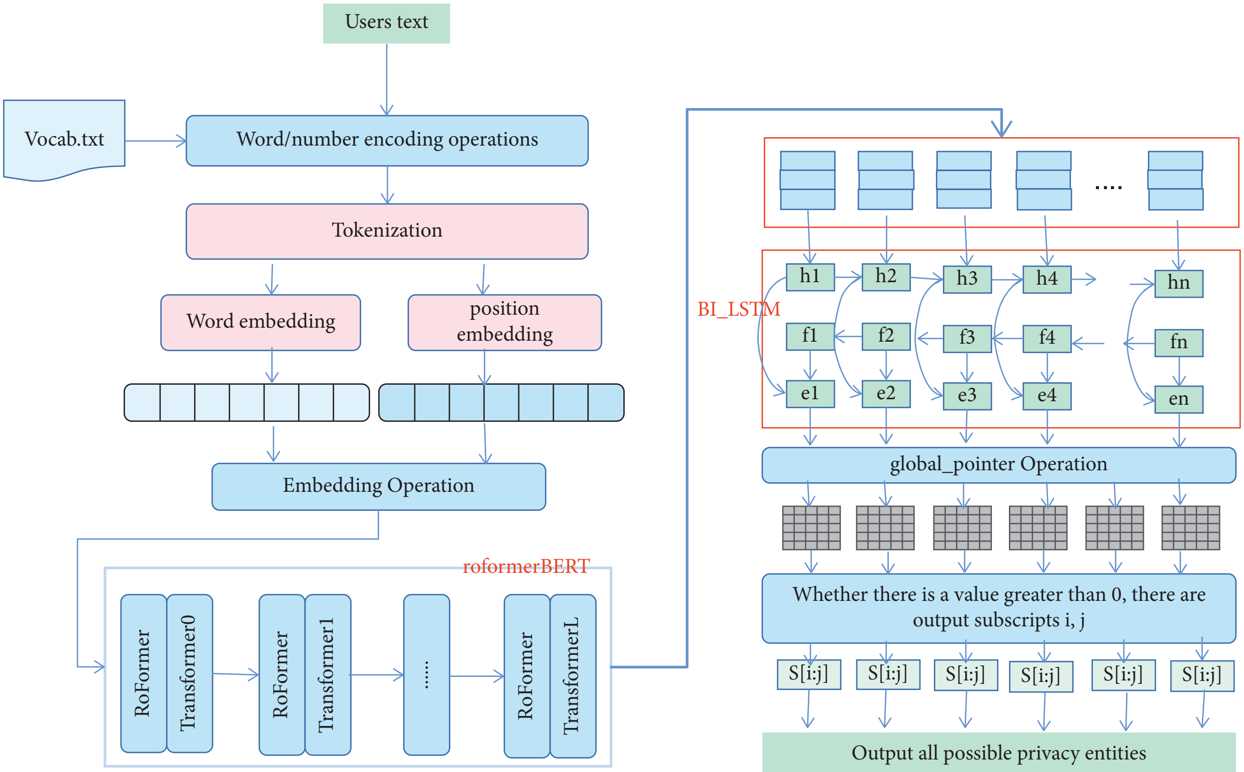

Figure 2: An overview of the DPER model. The input to this model includes text data, which are transformed into a data format by tokenization operation. Features are extracted using the roformerBERT model and the BI_LSTM model. Finally, the GP algorithm is used to predict the privacy entities. The parameter n indicates the length of the input sentence. The output $S\ [i\colon j]$ indicates that a privacy entity appears from the ith to the jth position in the input text.

privacy entities. The overall flow diagram of the DPER model is shown in Figure 2.

The specific steps of the privacy entity recognition model are as follows.

Step 1. The lexical text *Vocab.txt* file coming from the BERT pretraining model, which is a text mapping of a word to a word number, is used to convert the words in the input text into the corresponding number. Then, tokenization operation is conducted to obtain the position embedding vector and the text embedding vector.

Step 2. The converted text-embedding and position-embedding vectors are fused by embedding to obtain the feature expression of the text data. This feature expression can fit into the input of the BERT pretrained model.

Step 3. The embedding-processed feature vectors are reencoded using the rotational encoding algorithm. The purpose of reencoding is to change absolute position coding to relative position coding, increasing the amount of input to the data.

Step 4. The transformed encoding vectors are processed by the BERT model to obtain the feature expression of the text data. A $(1 * L)$ number vector is learned through the BERT model to obtain a $(348 * L)$ matrix, which can better represent hidden features in the text.

Step 5. The data processed by the BERT model are imported to the BI_LSTM model for processing. Processing with the BI_LSTM model yields data with sequence feature information.

Step 6. The data obtained in the fifth step are passed on to the GP layer for calculation. The GP layer divides the input tensor into five matrix outputs. The final privacy entity category of the output is eventually determined by performing the calculation on each matrix.

3.2.1. BERT Pretraining Model.

The BERT model adopts the encoder unit of the multi-layer transformer to enable the multi-layer encoder to learn the pretraining model of general knowledge through pretraining tasks and to transfer the model to complete downstream tasks. The BERT model structure is mainly composed of multiple layers of the embedding layer, as shown in Figure 3. The embedding layer of BERT consists of three parts: segment embeddings, position embeddings, and token embeddings. The token embedding layer is the normal embedding layer. The segment embedding layer is used to handle the classification task of input sentence pairs. The position embedding layer is the position encoding of words in a sentence. Overall, the BERT model is a combination of multiple embedding layers and attention mechanisms. The embedding layer plus an attention mechanism is the

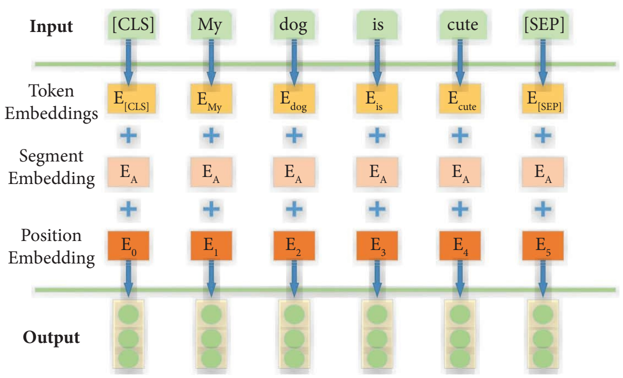

Figure 3: Embedding layer structure diagram.

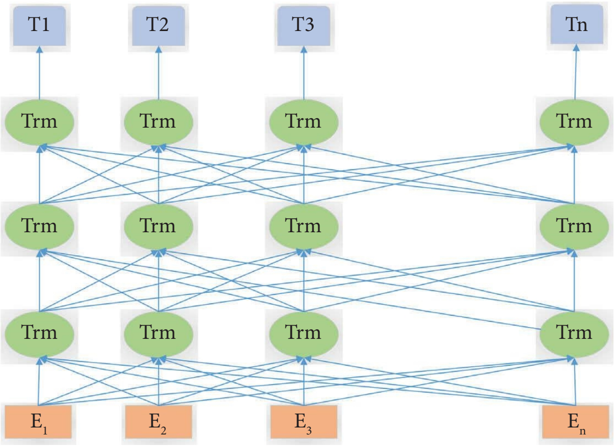

Figure 4: Structural diagram of BERT model.

transform model. As such, the BERT model is composed of multiple transforms. Figure 4 shows the structural diagram of the BERT model, where Trm represents the transform layer.

3.2.2. *roformerBERT Model.*

The *roformerBERT* model modifies the position embedding method of the BERT model. Rotary position embedding (RoPE) and the attention mechanism are used to realize the relative position embedding from the absolute position embedding [36]. The steps of RoPE are as follows.

(1) Absolute position information is added to q and k through formula (3) operation:

$$\widetilde{q}_m = f(q, m) \quad \widetilde{k}_n = f(k, n), \tag{3}$$

where f is an operation that indicates that q and k will have the absolute position information of m and n after f operation. m and n indicate absolute location information.

(2) By using the idea of the inner product calculation in the attention mechanism and the conjugate calculation of the complex numbers, the inner product is transformed into a form that can only be dependent on the relative position m-n [43, 58]. In this manner, absolute and relative positions are skillfully fused together, as shown in the following formula:

$$< \widetilde{q_m} e^{\mathrm{i}m\theta}, \widetilde{k}_n e^{\mathrm{i}n\theta} > \; = \mathrm{Re}\left[\widetilde{q_m} e^{\mathrm{i}m\theta} \widetilde{k}_n e^{\mathrm{i}n\theta}\right] = \mathrm{Re}\left[\widetilde{q_m}\widetilde{k}_n * e^{i(m-n)\theta}\right], \tag{4}$$

where Re[] denotes considering the real part of the result. $e^{\mathrm{i}m\theta}$ and $e^{\mathrm{i}n\theta}$ are representations that add imaginary parts to $\widetilde{q_m}$ and $\widetilde{k}_n$ for calculation, respectively. $*$ denotes conjugate calculation.

3.2.3. *GP Algorithm.*

The GP algorithm uses global normalization ideas to conduct entity recognition. It can recognize nested and nonnested entities without distinction [59]. The GP algorithm works better than the conditional random field (CRF) in nonnested (Flat NER) cases. It also yields better results in nested (nested NER) cases. The specific algorithm idea is presented in Algorithm 1. The mathematical calculation expression of the GP algorithm is provided as formula (5). f_{QDense} represents the fully connected operation for computing the query matrix, and formula (6) is its calculation procedure. $f_{K\mathrm{Dense}}$ represents the fully connected operation for computing the key matrix, and formula (7) is the calculation procedure. f_{Rope} represents the RoPE rotary encoding operation, and formula (8) is its calculation procedure. g_{BL} represents the BI_LSTM model operation.

$$H_{\mathrm{GP}} = f_{\mathrm{Rope}}\left(f_{\mathrm{QDense}}(H_{\mathrm{BL}}) \otimes f_{K\mathrm{Dense}}(H_{\mathrm{BL}})\right), \tag{5}$$

$$f_{\mathrm{QDense}}(H) = w_Q H + b_Q, \tag{6}$$

$$f_{K\mathrm{Dense}}(H) = w_k H + b_k, \tag{7}$$

$$f_{\mathrm{Rope}}(Q \otimes K) = \mathrm{Re}\left[Q^T * K * e^{(m-n)\theta}\right]. \tag{8}$$

3.2.4. *DPER Model Loss Function.*

Because there are too much nonentity data in the private entity identification dataset, the long tail phenomenon exists in the dataset. In this study, formula (9) is used to calculate loss. This calculation method improves the multi-label cross-entropy loss function, such that the score of the private tag class is higher than that of the nonprivate tag class [59, 60]. The inferential details of the formula are described in Appendix A.

$$\mathrm{loss} = \log\left(1 + \sum_{j \in P} e^{-S_j}\right) + \log\left(1 + \sum_{i \in N} e^{s_i}\right), \tag{9}$$

where P is the set of privacy class, N is the set of nonentity class, and S_i, S_j represent the category score.

3.3. *IPI Model.*

The IPI model is composed of the roformerBERT pretrained model and the UniLM model. The overall model diagram is presented in Figure 5. Formula (10) is a feature representation of the IPI model that uses the

```
Input: Attention mechanism head number, heads, the size of each head, head_size, and the input data, inputs.
Output: (inputs.shape[0], heads, inputs.shape[1], inputs.shape[1])type of tensor
(1): inputs $\Longleftarrow$ dense (inputs) #The dense is a Dense operation
(2): inputs $\Longleftarrow$ split(inputs, self.heads, axis = $-1$) #The split is a tangection function
(3): inputs $\Longleftarrow$ Keras.stack(inputs, axis = $-2$)
(4): qw $\Longleftarrow$ inputs[..., : head$_s$ize]
(5): kw $\Longleftarrow$ inputs[..., head$_s$ize:]
(6): qw, kw $\Longleftarrow$ RoPE (qw, kw) #RoPE rotary encoding
(7): logits $\Longleftarrow$ qw $\times$ kw #Calculate the internal product
(8): logits $\Longleftarrow$ sequence_masking (logits, mask) #exclude the padding mask as a mask
(9): mask $\Longleftarrow$ The lower triangle matrix of the logits was calculated
(10): logits $\Longleftarrow$ logits $-$ (1 $-$ mask) $*$ $e^{12}$
(11): Return logits $\Longleftarrow$ logits/self.head_size $**$ 0.5
```

Algorithm 1: Global_pointer algorithm.

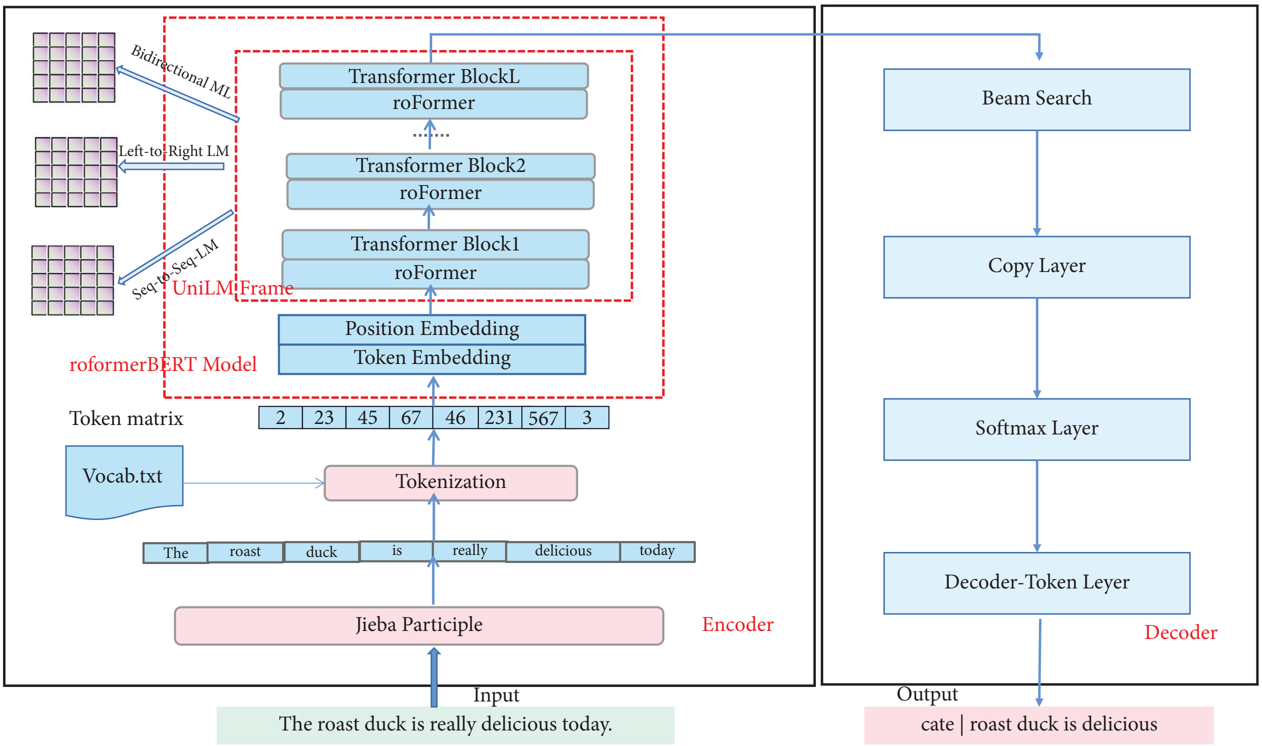

Figure 5: An overview of the IPI model. This model is the same as the traditional seq2seq framework, which is divided into encoding and decoding operations. In the encoding operations, the roformerBERT model is used for feature extraction, and inference data are generated through the seq2seq LM model in the UniLM framework. The beam search algorithm, copy operation, and softmax function are used in the decoding operation.

seq2seq mode for privacy inference. g_D and g_E represent the decoding and encoding operations, respectively. g_{BS} indicates that the data are processed using the beam search algorithm. g_{RFB} represents the roformerBERT model operation that contains rotational position encoding operations. H_{IPI} represents the output of the IPI model. Moreover, the model adopts the softmax function for normalization processing before token generation. The loss function used in this model is still the traditional cross-entropy loss function.

$$H_{\text{IPI}} = g_D\left(g_{\text{BS}}\left(g_E\left(g_U\left(g_{\text{RFB}}\left(X_{\text{in}}\right)\right)\right)\right)\right). \tag{10}$$

In the IPI model, we employ the roformerBERT model to extract text features. The UniLM framework is utilized to address the issue of BERT's inability to generate text, enabling the completion of unidirectional, sequence-to-sequence, and bidirectional prediction tasks, while integrating the advantages of autoregressive and autoencoder language models [61].

The specific steps of the IPI model are as follows.

Step 1. Word segmentation is performed on the input text data. The token dictionary adopts the word vector during roformerBERT pretraining, refining the characteristics of the input text more precisely. This model uses the Jieba word segmentation technique for word segmentation.

Step 2. The word encoding vector is obtained by converting the split text by using the word dictionary *Vocab.txt* file, which is a word-to-word text mapping. This current encoding conversion requires generating position encoding and word encoding.

Step 3. The encoding vector generated in the previous step is inputted into the roformerBERT + UniLM model for data generation and encoding. One token is outputted at a time.

Step 4. The beam search algorithm is used for text decoding (as shown in Algorithm 2). Step 3 of the loop selects the first n maximum-scored token of each output. The selected token is calculated with the previously generated token coding sequence. The sequence with the highest final score is selected to enter the next cycle.

Step 5. Determining whether the output value contains an end character flag or if the output string exceeds the predefined maximum length. Once these conditions occur during the loop, the output token sequence is converted into text output.

3.3.1. UniLM Model.

The UniLM framework is composed of multi-layer transformer networks, where the core is a BERT model. By converting the BERT model, the three tasks of bidirectional LM, left-to-right LM, and seq-to-seq LM can be completed simultaneously. Figure 6 shows the structural diagram of the UniLM framework [61]. In this study, the core network consists of 12 or 6 layers of the transformers. First, the input vector x_i is converted into $H_0 = [x_1, \ldots, x_{|X|}]$. Then, it is sent to the 12-layer or 6-layer transformer network. Each layer coding output is shown in formula (11). H^l represents the l-layer output.

$$H^l = \text{Transformer}_l\left(H^{l-1}\right). \tag{11}$$

Each layer controls the range of attention of each word by the mask matrix M. If an element in the matrix M has a value of zero, then it indicates attention; otherwise, it indicates no attention, and the corresponding feature is masked. Formula (13) is the calculation method of the mask matrix M. For the l-layer transformer, the output of the self-attention head A_l is calculated as shown in formula (14). Formula (12) is used to calculate the Q, K and V matrices.

$$Q = H^{l-1}w_l^Q, K = H^{l-1}w_l^K, V = H^{l-1}w_l^V, \tag{12}$$

$$M_{i,j} = \begin{cases} 0, & \text{Allow to attend,} \\ -\infty, & \text{Prevent from attending,} \end{cases} \tag{13}$$

$$A_l = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V_l. \tag{14}$$

In the IPI model, we choose the seq2seq ML model for the inference of interest privacy and the generation of the corresponding interpretable text. The seq2seq ML mode is a combination of bidirectional LM and left-to-right LM. Specifically, we define the input statement as $X = (x_1, x_2, \ldots, x_n)$ and the output statement as $Y = (y_1, y_2, \ldots, y_n)$. During model calculation, bidirectional LM operation is performed on X, while left-to-right LM operation is performed on Y. The calculation formula for the bidirectional LM operation is presented in (15), where the dimensions of H are the same as those of input X. g_{bert} representation is a BERT operation and $g_{\text{embedding}}$ indicates that X vector (matrix) is embedded after linear changes. Left-to-right LM (16) operation generates Y unidirectionally on the basis of the feature vector H given by the bidirectional LM operation. Formula (16) is mainly intended to calculate P (.). The sequence of Y is generated in the case of the maximum value. By using the seq2seq ML model, we can finally predict the interest attribute of the text and extract which data in the input text support this prediction.

$$H = g_{\text{bert}}\left(g_{\text{embedding}}(Xw + b)\right), \tag{15}$$

$$\text{argmax}\, P\left(\frac{Y}{H}\right) = P\left(\frac{y_1}{H}\right)P\left(\frac{y_2}{H} \cdot y_1\right) \ldots P\left(\frac{y_n}{H} \cdot y_1 \cdot \ldots \cdot y_{n-1}\right). \tag{16}$$

4. Experimental Evaluation

4.1. Dataset.

DL models involve numerous data for training. However, social network privacy datasets are extremely rare. Consequently, this work involves the construction of new datasets to train DL models. Among the large number of social networks, Sina Weibo is the most popular user platform and the most widely used social network platform in China, with 211 million daily active users. It searches and organizes the Sina Weibo corpus, and two datasets are used to train the DPER and IPI models. In the collection of private datasets, we use formula (17) as the collection standard of privacy statements. As long as an entity that can be identified as a natural person is present, along with his/her interests, address (LOC), job (JOB), educational background (EDU), and employee/company (COM), we mark the sentence as containing privacy information.

$$(\exists \text{Person} \vee \exists \text{BI}) \wedge (\exists \text{interest} \vee \exists \text{LOC} \vee \exists \text{JOB} \vee \exists \text{EDU} \vee \exists \text{COM}) \longrightarrow \text{Privacy}(\text{message}). \tag{17}$$

```
Input: Text dataset input, number of candidate sets topk, minimum distance min _ends, first symbol start_id, terminate the character end_id, The maximum length is generated maxlen.
Output: Generate the corresponding text
output_ids⟸start_id, output_scores⟸ [0]
for step = 0 to maxlen do
    scores, states⟸ predict(inputs, output_ids, states)
    if step == 0 then
        inputs⟸[inputs, inputs, inputs]
    end if
scores⟸output_scores + scores
indices⟸ The subscript of the largest topk values in scores is extracted.
output_ids⟸output_ids and indices merge into an array
output_scores⟸ Pick out the subscript value corresponding to the value of indices in scores
    if output_ids = = end_id then
        end_counts⟸ 1
    end if
    if output$_i$ds.shape[1] ≥ minlen then
        best_one⟸ Maximum value in the output_scores
        if end_counts = = 1 then
        Return output_ids[best_one]
        end if
    end if
end for
$x$⟸ Maximum value in the output_scores
Return output_ids[$x$]
```

Algorithm 2: Beam search algorithm.

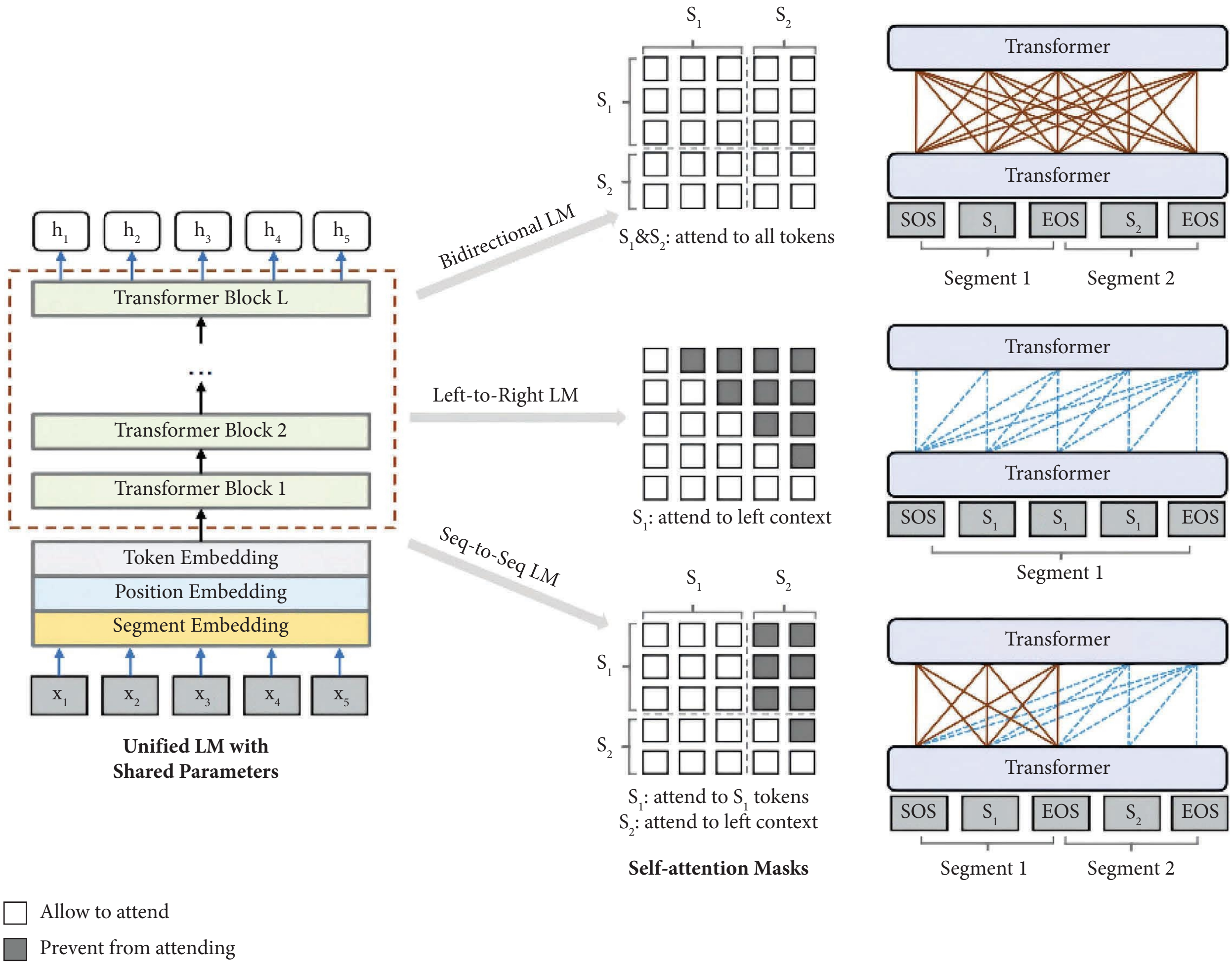

Figure 6: Structural diagram of UniLM [61].

Table 2: Distribution of the number of statements in the privacy entity recognition dataset.

Privacy entity items	Number of branches
BI	983
LOC	1575
JOB	2988
EDU	1320
COM	2148
All	9014

4.1.1. Privacy Entity Recognition Dataset. This dataset extracts the privacy data from the Sina Weibo dataset and eventually obtains 9,014 datasets with privacy entities. In this study, privacy entity mostly includes personal information (e.g., name, birthday, height, weight, etc.), address (e.g., current location, company location, home address, etc.), job, educational background, and employee/company. The specific distribution is presented in Table 2. Given that multiple privacy entities may be involved in one piece of data, the sum of the entity number of each privacy item in Table 2 is greater than the overall number of dataset entities.

Figure 7 calculates the specific number of each privacy entity item, including 2,128 privacy data items for LOC, 1,502 privacy data items for BI, 7,036 privacy data items for JOB, 2,883 privacy data items for EDU, and 4,002 privacy data items for COM.

4.1.2. Interest Privacy Inference Dataset. In this work, 4,694 interest datasets are retrieved using a data crawler technology in the interest region of Sina Weibo. These datasets have 12 interest categories, and the distribution of each category is shown in Figure 8. Among these datasets, 460 belong to the lifestyle category, 394 to the design aesthetics category, 391 to the games category, 536 to the sports category, 280 to the variety show category, 280 to the film and television category, 382 to the finance category, 346 to the tourism category, 260 to the mother-and-child category, 409 to the animation category, 442 to the reading category, and 514 to the food category. For each type, we mark its corresponding recognition basis.

4.2. Metrics. The evaluation indexes used in this study are precision (P) calculated using formula (18), recall (R) calculated using formula (19), $F1$ calculated using formula (20), and accuracy (ACC) calculated using formula (21). The specific calculation formulas of these evaluation indexes are as follows:

$$\text{Precision} = B\left(\text{TP}_j, \text{FP}_j, \text{TN}_j, \text{FN}_j\right)\frac{\text{TP}_j}{\text{TP}_j + \text{FP}_j}, \tag{18}$$

$$\text{Recall} = B\left(\text{TP}_j, \text{FP}_j, \text{TN}_j, \text{FN}_j\right)\frac{\text{TP}_j}{\text{TP}_j + \text{FN}_j}, \tag{19}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{20}$$

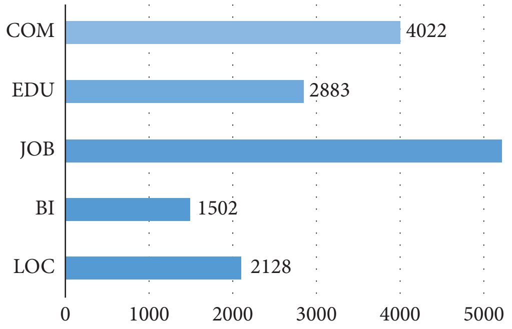

Figure 7: Distribution of the number of privacy items in the privacy recognition dataset.

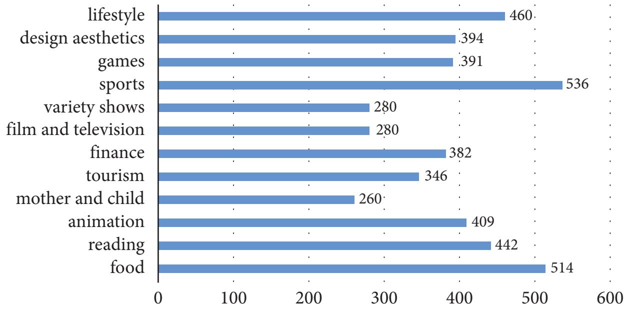

Figure 8: Interest privacy inference dataset distribution.

$$\text{Accuracy} = \frac{\text{TP}_j + \text{TN}_j}{\text{TP}_j + \text{FP}_j + \text{TN}_j + \text{FN}_j}, \tag{21}$$

where the true positives (TP_j) are the positive events that are correctly predicted, the true negatives (TN_j) are the negative events that are correctly predicted, the false positives (FP_j) are the negative events that are incorrectly predicted to be positive, and the false negatives (FN_j) are the positive events that are incorrectly predicted to be negative. j represents the corresponding category.

In this study, the generative text algorithm is adopted in the IPI model. Therefore, the more popular recall-oriented understudy for gisting evaluation (ROUGE) measure is used to test the effect of the generative text. ROUGE was presented in 2004 by I Chin-Yew Lin. It is a set of metrics for evaluating automatic summarization generation tasks and machine translation tasks [62]. The main ROUGE metrics are as follows: *rouge-1* (formula (22)), *rouge-2* (formula (23)), *rouge-L* (formula (24)), and *main* (formula (25)) [63].

The denominator in the *rouge-1* and *rouge-2* indicator formulas is the number of *n-gram* in the standard generated text, and the molecule is the number of *n-gram*, where the model-generated text and the standard generated text coincide. In the formula, gram_1 means *1-gram*, and gram_2 means *2-gram*. In the *rouge-L* index formula, LCS (X, Y) indicates the length of the longest common subsequence in the X and Y sequence. X represents the standard-generated text, Y represents the model-generated text, m and n indicate the length of X and Y, and β is a regulator. The *main* index is a weighted sum of the above three aforementioned indexes.

$$\text{rouge} - 1 = \frac{\sum_{S\in \text{ReferemceSummaries}} \sum_{\text{gram\_1}\in S} \text{Count}_{\text{match}}(\text{gram\_1})}{\sum_{S\in \text{ReferemceSummaries}} \sum_{\text{gram\_1}} \text{Count}(\text{gram\_1})}, \tag{22}$$

$$\text{rouge} - 2 = \frac{\sum_{S\in \text{ReferemceSummaries}} \sum_{\text{gram\_2}\in S} \text{Count}_{\text{match}}(\text{gram\_2})}{\sum_{S\in \text{ReferemceSummaries}} \sum_{\text{gram\_2}} \text{Count}(\text{gram\_2})}, \tag{23}$$

$$\text{rouge} - L = \frac{\left(1+\beta^2\right)(\text{LCS}(\text{XY})/m)(\text{LCS}(\text{XY})/n)}{(\text{LCS}(\text{XY})/m) + (\text{LCS}(\text{XY})/n)}, \tag{24}$$

$$\text{main} = 0.2 * \text{rouge} - 1 + 0.4 * \text{rouge} - 2 + 0.4 * \text{rouge} - L. \tag{25}$$

4.3. Ablation Experiment

4.3.1. Selection of Hyperparameters of the DPER Model. The DPER adopts the roformerBERT + BI_LSTM + GP structure in which the major hyperparameters include batch size, cycle number, and learning rate. Batch size and cycle number, which are set as 16 and 30 in this research, respectively, affect the training speed of the model. The learning rate is decisive for the final effect of the model, and this work reduces the learning rate from $1e-1$ to $1e-10$ by the order of the magnitude step of 0.1. During training, we discover that when the learning rate is greater than or equal to $1e-3$ and less than or equal to $1e-7$, gradient explosion occurs in the entire model training. Therefore, we conducted a test between $1e-3$ and $1e-7$, and the result of the learning rate training is presented in Figures 9 and 10. When the learning rate is $1e-4$, the optimal $F1$ of the training model is 96.72%. When the learning rate is $1e-5$, the optimal $F1$ of the training model is 98.83%. When the learning rate is $1e-6$, the optimal $F1$ of the training model is 81.68%. Hence, the model learning rate of the experiment is $1e-5$.

Each cycle during the training session cut the overall data into 620 pieces, and all of the trained models undergo 18,600 training sessions. A validation test is performed after the end of each cycle, and the results of the validation test are shown in Figure 11. When the learning rate of the model is $1e-5$, the $F1$ value of the validation set is the highest, and the model achieves the best result.

4.3.2. Selection of Hyperparameters of the IPI Model. The IPI model uses the principle of seq2seq model for model construction, and roformerBERT + UniLM is used to build the model. The major hyperparameters in the structure are the same: batch size, cycle number, and learning rate. Batch size and cycle number are 8 and 50, respectively. The learning rate plays a decisive role in the final effect of the model. The learning rate is screened from $1e-1$ to $1e-10$. In the experiment, the loss value is 0 when the learning rate is greater than or equal to $1e-3$ and less than or equal to $1e-7$. Therefore, we demonstrate the training situation from $1e-4$ to $1e-6$, and the training results of the specific learning rate are presented in Figure 12. Given that the seq2seq model structure is used, the encoder is trained during the training. To measure the effect of each text generation, we use the training set to test it. The quality of the generated text is measured using the ROUGE detection method. As shown in Figure 12, $1e-4$ can achieve the best results in all the four indicators, with the *main* index reaching 97.63%, the *rouge-1* index reaching 98.36%, the *rouge-2* index reaching 96.55%, and the *rouge-L* index reaching 98.36%.

4.4. Effects of the Models Developed in This Study

4.4.1. Effect of the DPER Model. This study constructs a test set to evaluate the final DPER model, with 560 pieces of data. The test set contains 1,123 privacy entities, including 456 LOC, 188 BI, 204 EDU, 181 JOB, and 90 COM privacy entities. The model is evaluated in terms of ACC, $F1$, P, and R. Tables 3 and 4 present these aspects. The DPER model is evaluated using four indicators: ACC, $F1$, P, and R. Tables 3 and 4 indicate the prediction effects of the DPER model. From the overall performance of the test set, ACC reached 91.80%, the $F1$ was 93.74%, the P value was 97.41%, and the R value was 90.33%. From the recognition of each privacy item, BI and EDU privacy entities are not as effective as the four other privacy indicators. The primary reason is that the sample space of the privacy entities marked by BI and EDU is relatively large, and the regularity is relatively complex, leading to the imperfect feature information learned by the model in this respect.

4.4.2. Effect of the IPI Model. A total of 1,200 interest test texts are collected to test the IPI model. Figure 13 shows the inference accuracy of each interest. The lowest accuracy of the model can reach 96% in interest inference, and some interest inferences can reach up to 100%. This outcome shows that the IPI model designed in this study is feasible for IPI.

4.5. Comparison of BERT Models per Version. With the extensive use of the BERT pretraining model in DL, various versions have also been produced accordingly. The 12-layer BERT, 6-layer BERT, 12-layer roformerBERT, and 6-layer roformerBERT models are compared. The model proposed here will be eventually run on the user clients, but some clients do not have sufficient memory, and thus, a relatively small 6-layer model is used for training. The BERT model can only handle 512 characters, and thus the roformerBERT model can extend the data processing length via rotary encoding. Table 5

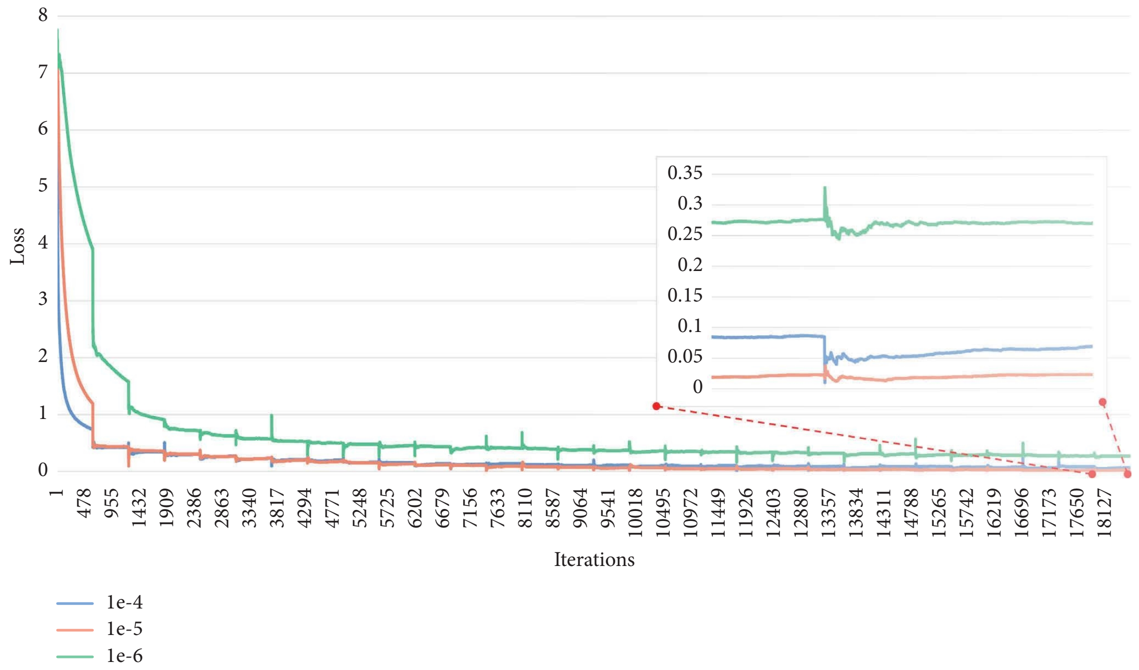

Figure 9: The loss value changes during the training of each learning rate for the DPER model.

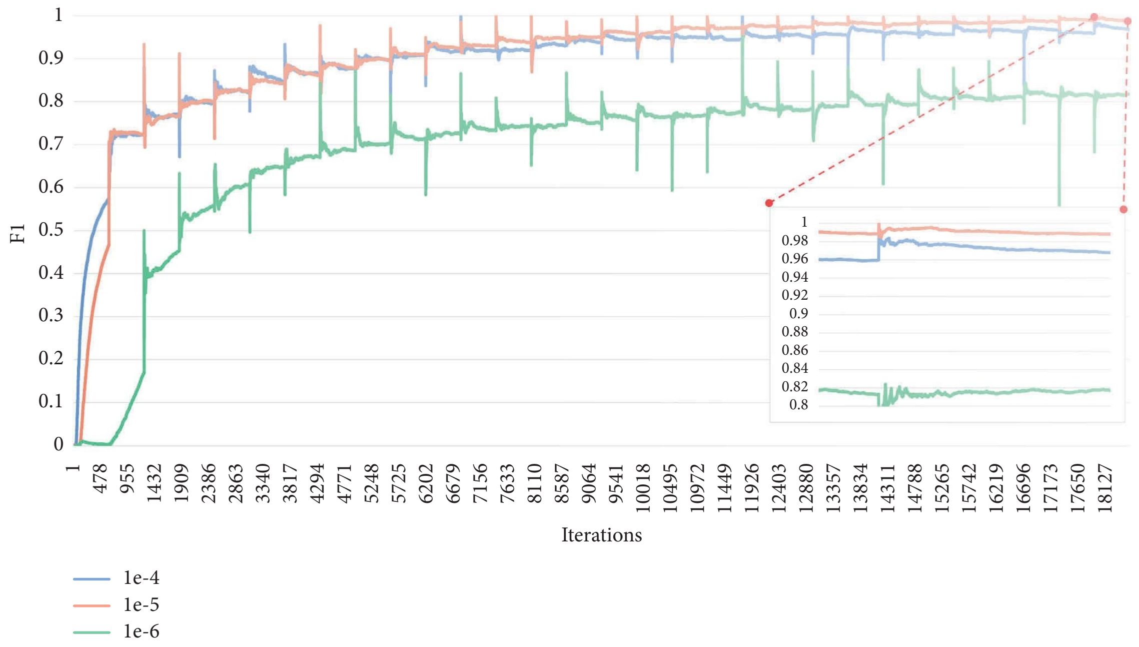

Figure 10: The $F1$ value changes during the training of each learning rate of the DPER model.

provides a comparison of each pretrained model of the privacy entity recognition model. The privacy recognition model built using the roformerBERT (12) pretraining model can achieve the highest effect. In the test, the $F1$ can reach 95.74%, P can reach 98.21%, and R can reach 92.53%. Simultaneously, the roformerBERT pretraining model exerts greater effect than the BERT pretraining model because in the roformerBERT model, rotation coding and data dictionary combined with words are used in token conversion. Figure 14 illustrates the recognition of each privacy entity of each version. As shown in Figure 14, the performance of the roformerBERT pretraining model is better than that of the BERT pretraining model. In Figure 14, however, the pretrained model of BERT (12) still performs better than the pretrained model of roformerBERT (6) in some indicators. The possible reason for such result is that the unbalanced distribution of privacy entities in the sample data leads to differences in model learning among privacy entities.

In the DPER model, we have proven that the roformer-BERT pretrained model outperforms the basic BERT model. Therefore, this study only uses the 6-layer roformerBERT and

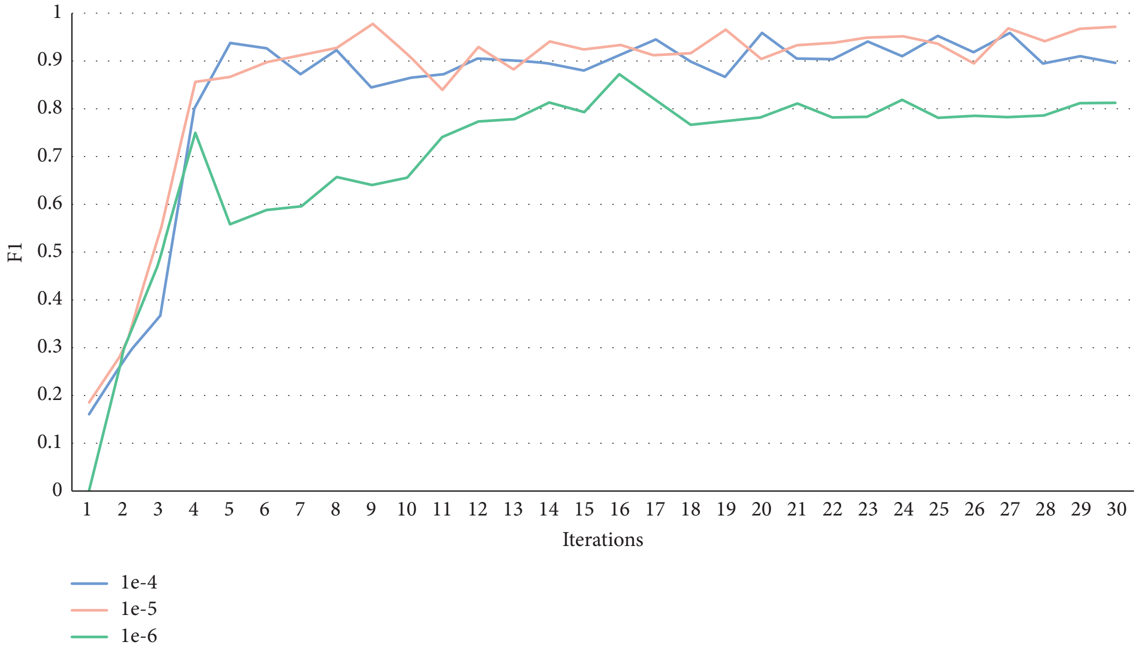

Figure 11: Results of the *F*1 values for each learning rate validation test.

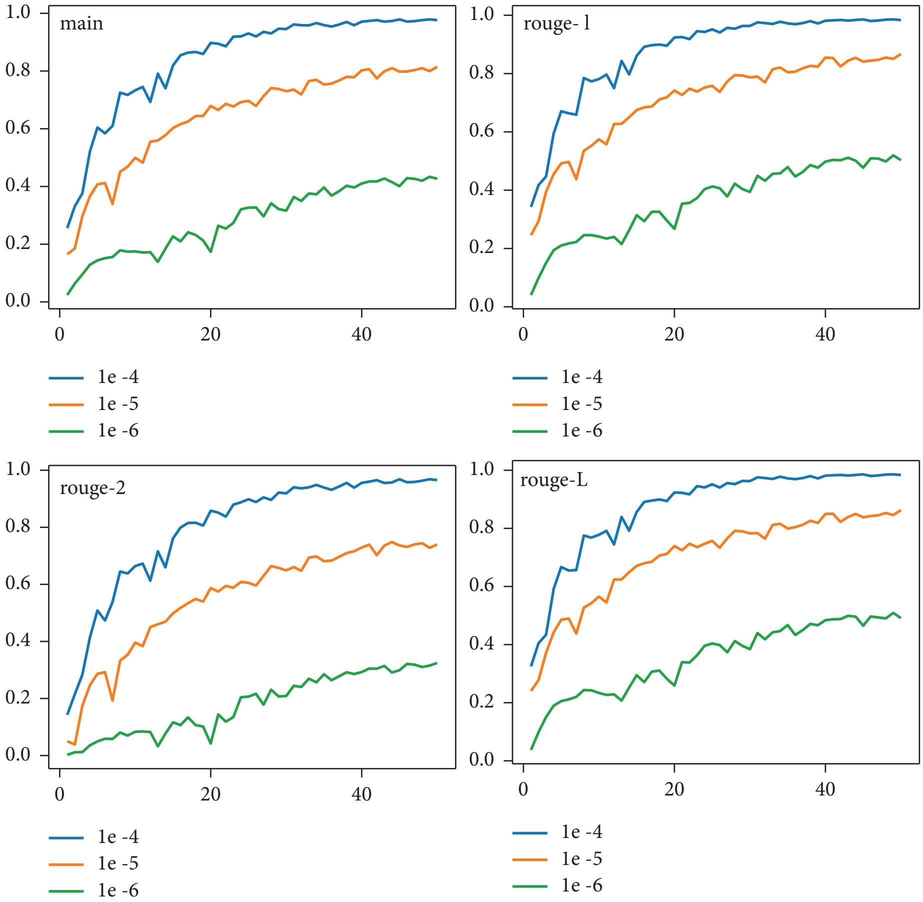

Figure 12: Results of the ROUGE indicators at each learning rate of the IPI model.

Table 3: Accuracy of the DPER model for the overall privacy entity and each privacy entity.

Privacy item	LOC	BI	EDU	JOB	COM	All
Predicted correct number	421	171	182	173	84	1084
Number of privacy entity	456	188	204	181	90	1123
Accuracy	92.51	90.95	89.21	95.58	93.33	91.80

Table 4: $F1$, P, and R of the DPER model for the overall privacy entity and each privacy entity.

Privacy item	LOC	BI	EDU	JOB	COM	All
$F1$	95.88	88.14	93.58	98.04	97.21	93.74
Precision (P)	98.77	90.95	96.56	96.68	96.67	97.41
Recall (R)	90.31	85.5	90.78	99.43	97.75	90.33

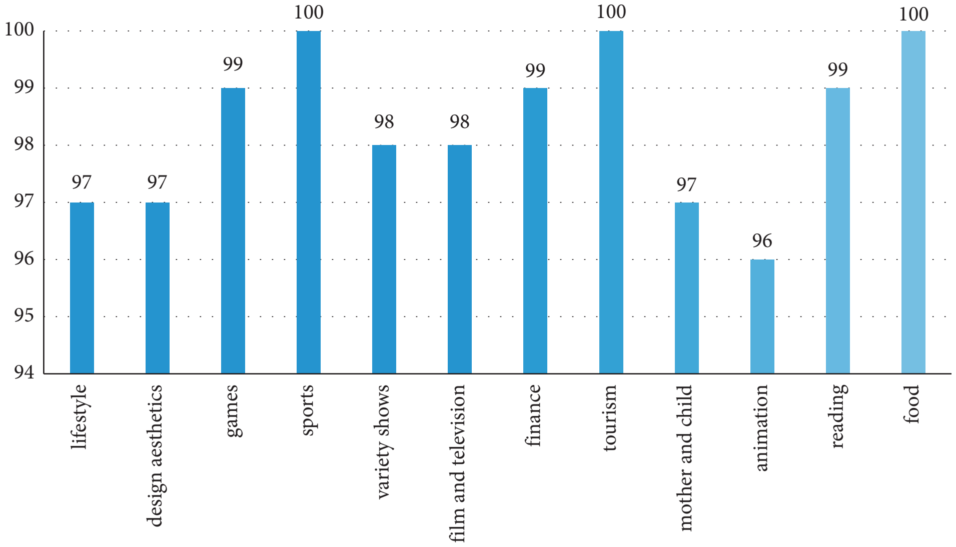

Figure 13: Prediction accuracy of the test dataset in the IPI model.

Table 5: Comparison of various pretrained models in privacy entity recognition models.

Model	$F1$	P	R
BERT(6)	93.22	93.45	92.99
BERT(12)	94.71	95.32	94.11
roformerBERT(6)	93.74	97.41	90.33
roformerBERT(12)	95.74	98.21	92.53

the 12-layer roformerBERT models in the IPI model. Specific pairs are shown in Figure 15. The blue lines in the figure denote the privacy inference test performance that uses the 6-layer roformerBERT pretrained model. The yellow lines represent the privacy inference performance of the 12-layer roformerBERT pretrained model. Overall, the results of using the two pretrained models for IPI are nearly the same. The *main* index can reach more than 97%, the *rouge-1* index can reach more than 98%, the *rouge-2* index can reach more than 96%, and the *rouge-L* index can reach more than 98%. However, the overall fluctuation of the model with the 12-layer roformerBERT is relatively large during training, probably because the overall parameter number of the pretraining model with the 12-layer roformerBERT is relatively large. Meanwhile, the amount of data we have inputted is relatively small, resulting in a large fluctuation during learning.

4.6. Comparison with Other Models. The privacy entity recognition model in this study is designed on the basis of the principle of entity recognition. This research makes a comparative analysis of several popular entity models. The models for comparison are the BI_LSTM-CRF model [64], BERT-CRF model [65], ALBERT-BI_LSTM-CRF model [66], EN2_BI_LSTM-CRF-CRF model [67], and ALBERT-MogAtt_BI_LSTM-CRF model [68]. P, R, and $F1$ are compared. The comparison results are provided in Table 6 and Figure 16. Table 6 indicates that the performance indexes of our model are higher than those of the BI_LSTM-CRF, BERT-CRF, ALBERT-BI_LSTM-CRF, EN2_BI_LSTM-CRF-CRF, and ALBERT-MogAtt_BI_LSTM-CRF models. All these models use CRF for the final physical output. Although this method can achieve good results in many domains, the composition of privacy entities is complex. Moreover, the

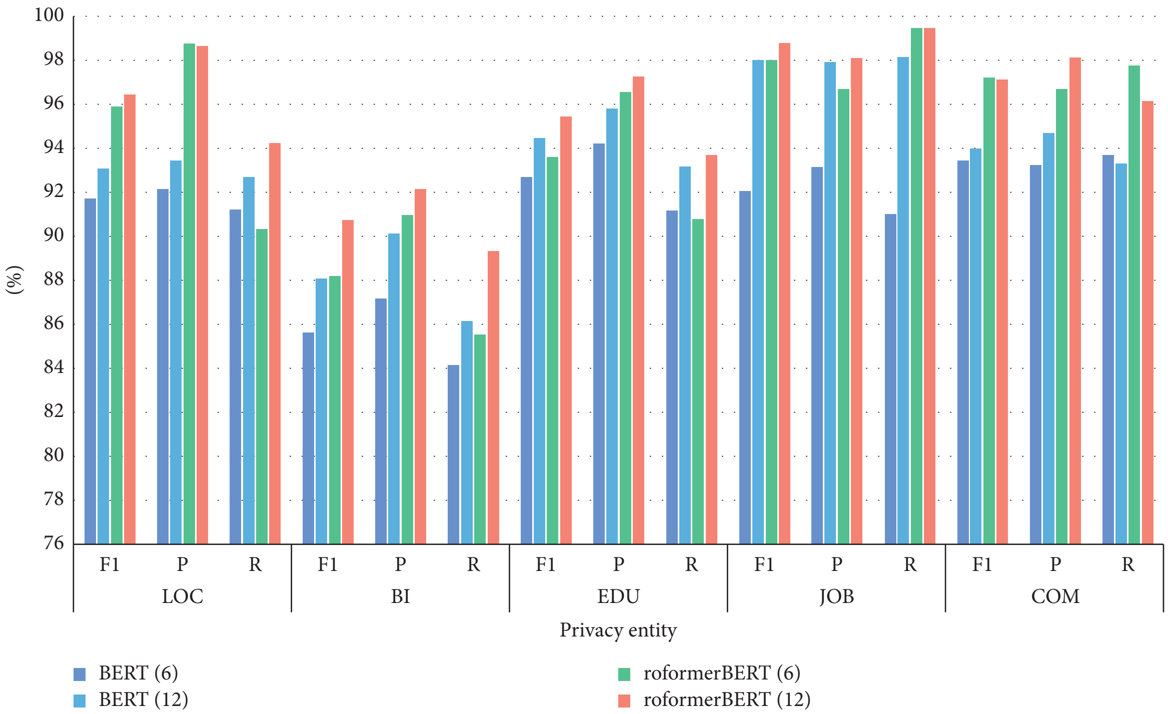

Figure 14: Performance of various BERT models in each privacy entity recognition.

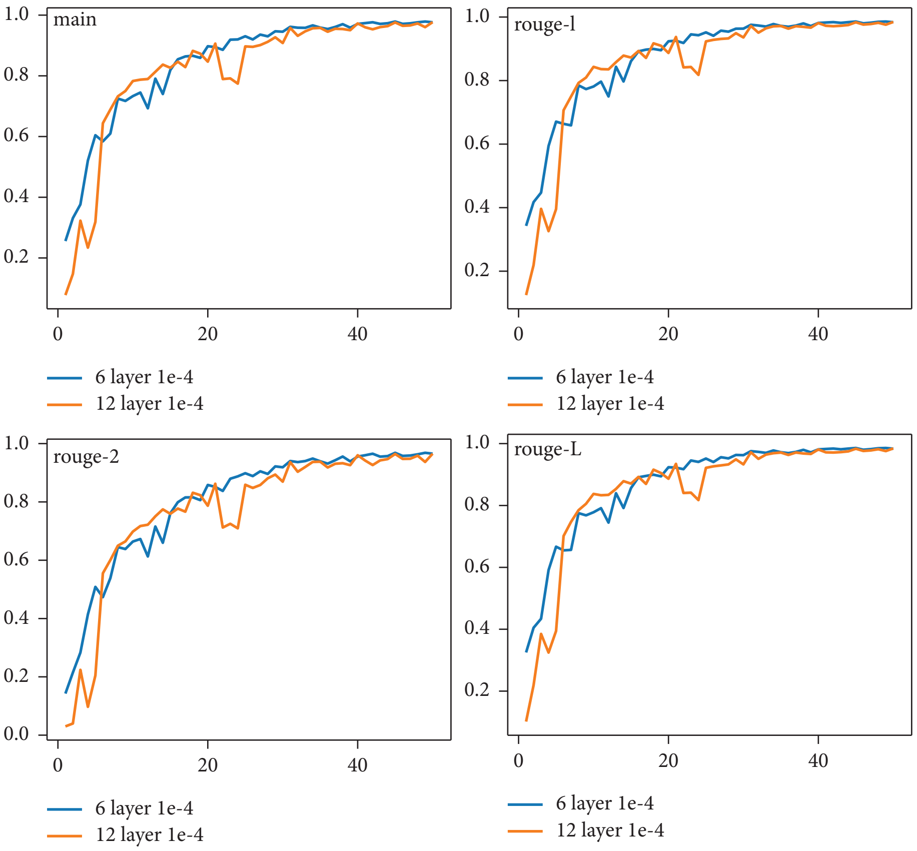

Figure 15: Comparing the impact of different layers of RoformerBERT on the IPI model.

Table 6: Comparison with the other entity recognition models.

Model	$F1$	P	R
BI_LSTM-CRF [64]	86.69	85.65	87.77
BERT-CRF [65]	91.62	91.11	92.13
ALBERT-BI_LSTM-CRF [66]	87.31	90.45	84.37
EN2_BI_LSTM-CRF-CRF [67]	89.86	91.06	88.7
ALBERT-MogAtt_BI_LSTM-CRF [68]	93.32	95.53	91.21
DPER model	95.74	98.21	92.53

model has many nested entities. Thus, the output of privacy entities by CRF is not very favorable. The method developed in the current research can effectively deal with these problems, achieving good results in privacy entity recognition. Figure 16 illustrates the recognition of each privacy entity in a model. From the $F1$ value, the current model is about 10% higher than the BI_LSTM-CRF model, about 4% higher than the BERT-CRF model, about 8% higher than the ALBERT-BI_LSTM-CRF model, about 5% higher than the EN2_BI_LSTM-CRF model, and about 2% higher than the ALBERT-MogAtt_BI_LSTM-CRF model. This outcome indicates that the privacy entity recognition model proposed in the current research is feasible.

This study also compares the proposed IPI model with the more popular interest recognition models at present, namely, the char2vec + CNN and word2vec + CNN models [38], as shown in Figure 17. The figure indicates that the interest inference effect of the proposed model is not different from those of the char2vec + CNN and word2vec + CNN models. Moreover, the inference effect of each interest item exhibits its own advantages and disadvantages. However, our model can provide the interpretability of the given interest inference, i.e., which information can be outputted to support the given interest inference result. Therefore, our IPI model is also better than existing models.

4.7. Complexity Analysis

4.7.1. Number of Parameters. The model parameters and test times of our proposed DPER and IPI models are provided in Table 7. For each model, we design a large model and a small model to meet the deployment requirements on different hardware platforms. The large model uses a 12-layer roformerBERT model, while the small model uses a 6-layer roformerBERT model. It can be seen from Table 7 that the number of DPER model parameters is 124 M and 30 M, which is 2.31 s and 1.17 s for the same sentence, respectively. The number of IPI model parameters is 102 M and 19 M, and 6.72 s and 3.05 s for the same sentence, respectively. The IPI model requires a long test time, primarily due to the large time consumption of generating text. However, it is still within the allowable range.

4.7.2. Model Complexity

(1) Time Complexity of DPER Model. The DPER model is composed of roformerBERT model, BI_LSTM model, and GP algorithm, and thus the overall time complexity of the model is sum of these three models. The following discussion specifically analyzes overall model time complexity.

(2) Time Complexity of the roformerBERT Model. The roformerBERT model is composed of an embedding layer, a position encoding layer, an attention layer, a dense layer, an add layer, a norm layer, and a feedforward fully connected layer. The time complexity of the embedding layer is $O(E_{L,V}^{\text{in}} * E_{V,H}^{\text{out}})$, where $E_{L,V}^{\text{in}}$ represents the input of the embedding layer, L represents the length of the input text, and V represents the dimension of the word dictionary. $E_{V,H}^{\text{out}}$ represents the output of the embedding layer, and H represents the dimension of the word vectors output by the roformerBERT model. The time complexity of the position encoding layer is $O(P_{1,L}^{\text{in}} * P_{L,H}^{\text{out}})$, where $P_{1,L}^{\text{in}}$ represents the input of the position encoding layer, and $P_{L,H}^{\text{out}}$ represents the output of the position encoding layer. The time complexity of attention layer is $C * O(A_{L,H}^{\text{in}} * A_{H,64}^{\text{out}})$, where C represents the number of layers in the attention layer, $A_{L,H}^{\text{in}}$ represents the input of attention layer, and $A_{H,64}^{\text{out}}$ represents the output of attention layer. The time complexity of dense layer is $O(C * L * H * H)$, where C represents the number of dense layers. The time complexity of add and norm layer is $C * [O(L * H) + \mathrm{O}(H * 2 * 2)]$. The time complexity of the feedforward fully connected layer is $C * O(L * H * I)$, where I represents the number of hidden neurons in the feedforward full link layer.

(3) Time Complexity of the BI_LSTM Model. According to the principle of the LSTM model, the time complexity of the LSTM model is $O(\widehat{H} * L + \widehat{H} * \widehat{H} * \widehat{H})$, where $\widehat{H}$ is the number of hidden layers of the model. This study uses BI_LSTM, which adopts a bidirectional LSTM model, and thus model complexity is $2 * O(\widehat{H} * L + \widehat{H} * \widehat{H} * \widehat{H})$.

(4) Time Complexity of the GP Algorithm. In this study, the GP algorithm is essentially a multi-head attention layer, with a time complexity similar to that of attention. Its time complexity is $n * O(G_{L,H}^{\text{in}} * G_{H,64}^{\text{out}})$, where $G_{L,H}^{\text{in}}$ is the input of the GP algorithm, $G_{H,64}^{\text{out}}$ is the output of the GP algorithm, and n is the number of categories of the entity.

In summary, the time complexity of the DPER model is $\begin{matrix} O(E_{L,V}^{\text{in}} * E_{V,H}^{\text{out}}) + O(P_{1,L}^{\text{in}} * P_{L,H}^{\text{out}}) + C * O(A_{L,H}^{\text{in}} * A_{H,64}^{\text{out}}) + O(C * L * H * H) + C * [O(L * H) + O(H * 2 * 2)] \\ +C * O(L * H * I) + 2 * O(\widehat{H} * L + \widehat{H} * \widehat{H} * \widehat{H}) + n * O(G_{L,H}^{\text{in}} * G_{H,64}^{\text{out}}) \end{matrix}$.

(5) Time Complexity of the IPI Model. The IPI model is designed in accordance with the seq2seq mode, and thus its time complexity is composed of the complexity of the encoder and decoder modules. The following is the complexity of the two modules.

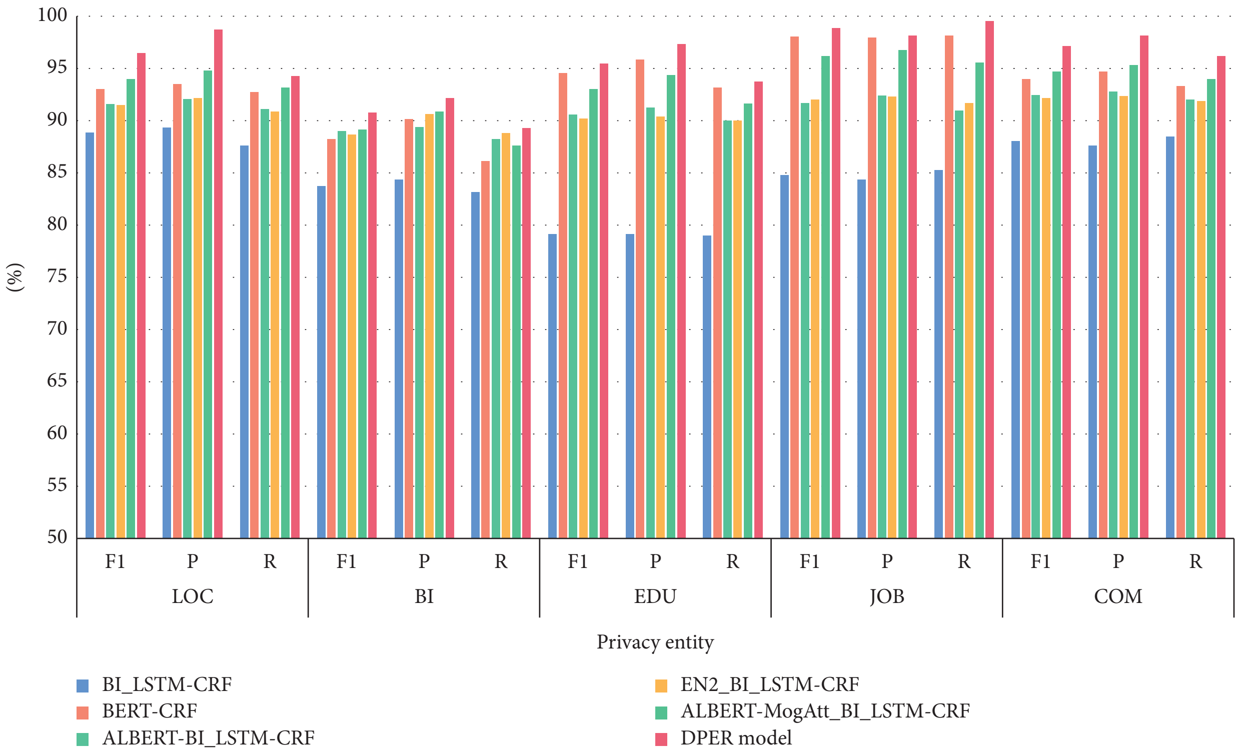

Figure 16: The recognition effect of each privacy entity recognition in each entity recognition model.

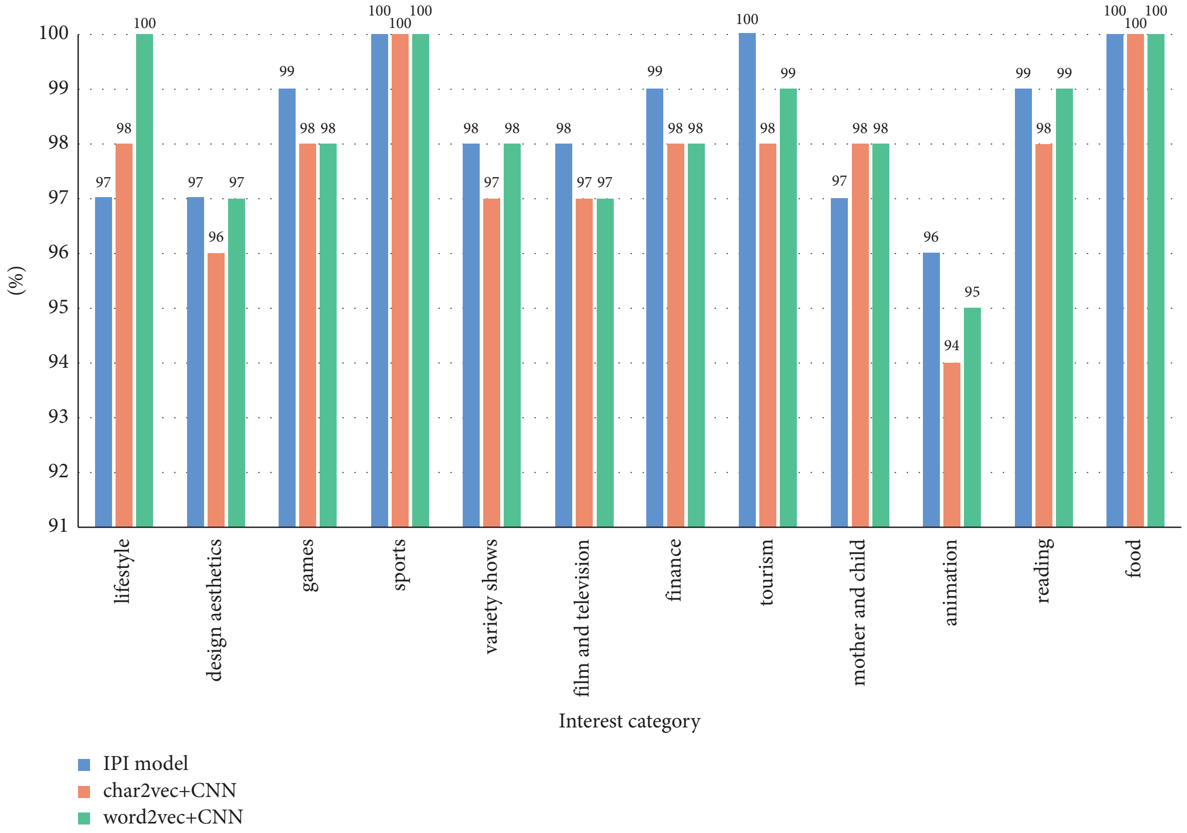

Figure 17: The accuracy comparison of the IPI model for each interest.

Table 7: Comparative analysis of the model parameters and the test time.

Model	Test time (s)	Model parameter
DPER model (BERT(12))	2.31	124,244,993
DPER model (BERT(6))	1.17	30,242,560
IPI model (BERT(12))	6.72	102,058,586
IPI model (BERT(6))	3.05	19,012,058

Table 8: Comparison of the DPER model with the traditional self-disclosure privacy identification model of social networks.

Sentence	Model	Output results
I moved to Chinatown today. Chinatown is really busy.	Traditional model [53, 54]	Privacy types: address
	Our model	Privacy types: address Location: moved to Chinatown

(1) Encoder module time complexity: The encoder module is composed of the UniLM framework and the roformerBERT model. UniLM framework only controls the number of masks of the text, without the actual complexity consumption, and the primary complexity consumption is the roformerBERT model. The roformerBERT model is introduced in detail in the preceding section; however, this module also adds the mask language model (MLM) task layer after the output of the roformerBERT model. The MLM task layer is composed of the MLM-dense layer, the MLM-norm layer, the MLM-bias layer, and the MLM-activation layer. The time complexity of the MLM-dense layer is $O(L*H*H)$, that of the MLM-norm layer is $O(L*H)+O(H*2*2)$, and that of the MLM-bias layer has no practical operation and only provides data. The time complexity of the MLM-activation layer is $O(M_{L,H}^{\text{in}}*M_{H,V}^{\text{out}})$, where $M_{L,H}^{\text{in}}$ represents the input of the model, L denotes the length of the text, and H represents the output word vector dimension of the roformerBERT model. $M_{H,V}^{\text{out}}$ represents the output of the model, and V denotes the dimension of the word dictionary.

(2) Decoder module time complexity: The decoder module is composed of beam search layer, copy layer, and softmax layer. The time complexity of the beam bearch layer is $O(N*T*K)$, where N represents the length of the data input from beam bearch layer, T represents the number of time steps to be decoded, and K represents the first K token for each selection. The softmax layer is mostly for normalizing the output of the beam search layer, and its time complexity is $O(N+T)$.

In summary, the overall time complexity of the IPI model is $\begin{array}{l} O(E_{L,V}^{\text{in}}*E_{V,H}^{\text{out}})+O(P_{1,L}^{\text{in}}*P_{L,H}^{\text{out}})+C*O(A_{L,H}^{\text{in}}*A_{H,64}^{\text{out}})+O(C*L*H*H)+C*[O(L*H)+O(H*2*2)] \\ +C*O(L*H*I)+O(L*H*H)+O(L*H)+O(H*2*2)+O(M_{L,H}^{\text{in}}*M_{H,V}^{\text{out}})+O(N*T*K)+O(N+T) \end{array}$.

4.8. Discussion. The DPER model is proposed to solve the problem, in which the traditional social network self-disclosure privacy identification model can only provide the type of privacy leakage, but not the corresponding location of the privacy leakage. Table 8 shows the difference between our method and the traditional self-disclosure privacy recognition model of social networks. As indicated in the table, our model enables users to more directly understand which words are revealing their privacy.

The DPER model uses NER to extract privacy, but the nested privacy entities cannot be extracted by the traditional CRF-based NER method. Therefore, the GP algorithm is used to perform the extraction of the nested privacy. For accuracy, the following statement is provided as an example: "imperceptibly come to Hainan Qianfan Culture Media Co. Ltd. has been more than a year." This sentence has nested address privacy, i.e., "Hainan" is an address and "Hainan Qianfan Culture Media Co. Ltd." is a company. Named entity identification based on CRF will directly identify the company entity of Hainan Qianfan Culture and Media Co. Ltd. but cannot identify the address entity of Hainan. The specific model prediction results are shown in Table 9. In Table 9, B_COM represents the beginning of a company entity, I_COM is the middle and end of a company entity, and O is not an entity. Table 8 indicates that the CRF-based models do not predict the "Hainan" entity. Our model differs from the traditional BIO output because it outputs a coordinate (i, j), where i represents the start position of an entity and j represents its end position. In Table 9, (6, 10) represents a company entity that starts from the sixth position in the sentence and extends up to the tenth position. In Table 9, the BLC, BC, ABLC, EBLCF, and ABLMBLC, respectively, refer to the BI_LSTM-CRF [64] model, BERT-CRF [65] model, ALBERT-BI_LSTM-CRF [66] model, En2_BI_LSTM-CRF [67] model, and ALBERT-MogAtt_BI_LSTM-CRF [68] model.

Given the low accuracy of extracting interest privacy by using the NER model, the reason may be the variety of interest expressions in the text. Although many interest recognition models are available to identify interest, these algorithms cannot provide the interpretability to support this interest. To solve this problem, we propose the IPI model for detecting interest. This model adopts the design model of seq2seq and the popular UniLM framework to transform the BERT model into a generative model and generate the explanatory text that supports the interest that exists in the source text. For accuracy, the following statement is provided as an example: "I made braised pig trotters at home, which are easy to make, with a chewy and soft texture and a delicious spicy and savory taste. They are full of collagen, so delicious!" Using the traditional interest classification algorithm, this statement will only indicate an interest in food. Using the IPI model, this statement can indicate an interest in food and provide the corresponding text to support this judgment. The specific output is shown in Table 10.

In summary, our proposed PA framework is fundamentally different from traditional privacy perception models. Our model can not only provide accurate types of privacy leakage but also offer a specific text description that supports this type of leakage.

Table 9: The DPER model-predicted result for sentence.

Sentence	BLC [64]	BC [65]	ABLC [66]	EBLCF [67]	ABLMBLC [68]	Our model
I have unconsciously been at Hainan Qianfan Culture Media Co., Ltd. for over a year now	o	o	o	o	o	Address: (6, 6)
	o	o	o	o	o	
	o	o	o	o	o	
	o	o	o	o	o	
	o	o	o	o	o	
	B_COM	B_COM	B_COM	B_COM	B_COM	
	I_COM	I_COM	I_COM	I_COM	I_COM	
	I_COM	I_COM	I_COM	I_COM	I_COM	
	I_COM	I_COM	I_COM	I_COM	I_COM	
	I_COM	I_COM	I_COM	I_COM	I_COM	Company: (6, 10)
	o	o	o	o	o	
	o	o	o	o	o	
	o	o	o	o	o	
	o	o	o	o	o	
	o	o	o	o	o	

Table 10: The IPI model-predicted result for sentence.

Sentence	Model	Output
I made braised pig trotters at home, which are easy to make, with a chewy and soft texture and a delicious spicy and savory taste. They are full of collagen, so delicious!	Interest identification model [38]	Interest: food
	IPI model	Interest: food Interpretation: braised pig trotters, a chewy and soft texture and a delicious spicy and savory taste, so delicious

5. Conclusion

This study proposes a PA framework for social networks that can automatically sense sensitive text information shared by users. It accurately locates which part of the text is leaking sensitive information and sends these privacy data as feedback to users to enhance their PA. This framework consists of two parts.

The first part is the direct privacy module, which uses named entities to extract a direct privacy entity. In this module, we combine the roformerBERT model, BI_LSTM model, and GP algorithm to train the DPER model that can not only identify private information in social text but also provide the location of private information. The model proposed in this module is tested on a test set, and it can reach 95.74%, 98.21%, and 92.53% in the indexes of $F1$ score, P, and R, respectively. The second part is the indirect privacy module. Some indirect privacy leaks are difficult to uncover in our experiment. This module combines the roformerBERT model and the UniLM framework to construct an IPI model for users. Meanwhile, interpretable text information is added when training the model. The designed IPI model can not only identify which privacy information of interest is being leaked in social text but also provide which information is serving as a guide in the corresponding text. The model in this module can reach the following indexes: *main* is 97.63%, *rouge-1* is 98.36%, *rouge-2* is 96.55%, and *rouge-L* is 98.36%. The proposed model framework can be applied to social network scenarios, text desensitization scenarios, privacy measurement calculations, and other scenarios. Simultaneously, we develop an application to provide users with privacy-aware services. Our application adopts a lightweight model. The data provided by users are only calculated locally, and no data collection is performed.

Although the model framework developed in this study can obtain good results in social network PA, it still requires considerable improvement to preserve personal privacy data in the entire social network. With regard to the definition of privacy, this study adopts a sweeping definition. However, the subject of privacy is humans, and different individuals have varying definition scopes of privacy. Designing a personal PA framework is the subject of our future research. Simultaneously, common privacy disclosure is the primary source of privacy disclosure in social networks, and the recognition of common privacy disclosure is another subject for future research.

Appendix

A. Derivation of the Formula of Loss Function

For the multi-label classification task, our goal is to make each target class score no less than that of each nontarget class. P is a label class, Q is a nonlabel class, and s represents the scores of each class. The loss value is calculated using the cross-entropy function as shown in the following formula:

$$\log\left(1+\sum_{i\in Q, j\in P} e^{S_i-S_j}\right)=\log\left(1+\sum_{i\in Q} e^{S_i}\sum_{j\in P} e^{-S_j}\right). \quad (A.1)$$

Make a class 0 so that the label class scores greater than S_0 and nonlabel class scores less than S_0. In order to satisfy $S_i < S_j$, it is necessary to add $e^{S_i-S_j}$ to the loss calculation formula. Overwrite formula (A.1) to get the following formula:

$$\log\left(1+\sum_{i\in Q, j\in P} e^{S_i-S_j}+\sum_{i\in Q} e^{S_i-S_0}+\sum_{j\in P} e^{S_0-S_j}\right). \quad (A.2)$$

Simplify formula (A.2):

$$\log\left(e^{S_0}+\sum_{i\in Q}e^{S_i}\right)+\log\left(e^{-S_0}+\sum_{j\in P}e^{-S_j}\right). \tag{A.3}$$

Set $S_0 = 0$ to get

$$\log\left(1+\sum_{i\in Q}e^{S_i}\right)+\log\left(1+\sum_{j\in P}e^{-S_j}\right). \tag{A.4}$$

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Gan Liu was responsible for conceptualization, investigation, methodology, formal analysis, validation, and original draft preparation. Hui Li was responsible for supervision, methodology, project administration, and funding acquisition. Xiongtao Sun was responsible for supervision, formal analysis, validation, and original draft preparation. Yiran Li and Shuchang Zhao were responsible for review and editing and resources. Zhen Gou was responsible for supervision, project administration, and funding acquisition.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant nos. 61932015, U1836203, and 62262013.

References

[1] A. Shabtai, Y. Elovici, and L. Rokach, *A Survey of Data Leakage Detection and Prevention Solutions*, pp. 1–99, Springer, Berlin, Germany, 2012.

[2] B. Breve, L. Caruccio, S. Cirillo, D. Desiato, V. Deufemia, and G. Polese, *Enhancing User Awareness During Internet Browsing*, pp. 71–81, ITASEC, 2020.

[3] Y. Alsarkal, N. Zhang, and H. Xu, "Your privacy is your friend's privacy: examining interdependent information disclosure on online social networks," in *Proceedings of the 51st Hawaii International Conference on System Sciences*, pp. 1–10, Honolulu, HI, USA, January 2018.

[4] H. J. Nam, H. Y. Choi, H. J. Shin et al., "Security and privacy issues of fog computing," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 42, no. 1, pp. 257–267, 2017.

[5] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, "Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1317–1332, 2018.

[6] T. J. Holt and J. R. Lee, "A crime script model of Dark web Firearms Purchasing," *American Journal of Criminal Justice*, vol. 48, no. 2, pp. 509–529, 2022.

[7] L. Xu, C. Jiang, N. He, Z. Han, and A. Benslimane, "Trust-based collaborative privacy management in online social networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 48–60, 2019.

[8] Z. Feng, F. Cong, K. Chen, and Y. Yu, "An empirical study of user behaviors on pinterest social network," in *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pp. 402–409, IEEE, Atlanta, GA, USA, November 2013.

[9] E. Pallarès, D. Rebollo-Monedero, A. Rodríguez-Hoyos, J. Estrada-Jiménez, A. M. Mezher, and J. Forné, "Mathematically optimized, recursive prepartitioning strategies for k-anonymous microaggregation of large-scale datasets," *Expert Systems with Applications*, vol. 144, Article ID 113086, 2020.

[10] J. Wang, Z. Cai, and J. Yu, "Achieving personalized k-Anonymity-Based content privacy for autonomous vehicles in CPS," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4242–4251, 2020.

[11] S. Cheng, "Research on data privacy protection technology of social network users based on differential disturbance," *Ain Shams Engineering Journal*, vol. 13, no. 5, Article ID 101745, 2022.

[12] X. Chen, Z. Jiang, H. Li, J. Ma, and P. S. Yu, "Community hiding by link perturbation in social networks," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 3, pp. 704–715, 2021.

[13] C. Yan, Z. Ni, B. Cao, R. Lu, S. Wu, and Q. Zhang, "UMBRELLA: user demand privacy preserving framework based on association rules and differential privacy in social networks," *Science China Information Sciences*, vol. 62, no. 3, pp. 1–3, 2018.

[14] Z. Gao, Y. Huang, L. Zheng, H. Lu, B. Wu, and J. Zhang, "Protecting location privacy of users based on trajectory obfuscation in mobile crowdsensing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6290–6299, 2022.

[15] T. Zhu, J. Li, X. Hu, P. Xiong, and W. Zhou, "The dynamic privacy-preserving mechanisms for online dynamic social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 6, pp. 2962–2974, 2022.

[16] X. Yin, X. Hu, Y. Chen, X. Yuan, and B. Li, "Signed-PageRank: an efficient influence maximization framework for signed social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 2208–2222, 2021.

[17] S. Wen, M. S. Haghighi, C. Chen, Y. Xiang, W. Zhou, and W. Jia, "A sword with two edges: propagation studies on both positive and negative information in online social networks," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 640–653, 2015.

[18] T. Wang, Y. Mei, W. Jia, X. Zheng, and M. Xie, "Edge-based differenital privacy computing for sensor-cloud systems," *Journal of Parallel and Distributed Computing*, vol. 136, no. 3, pp. 540–551, 2019.

[19] M. Joseph, J. Mao, and A. Roth, "Exponential separations in local differential privacy," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, S. Chawla, Ed., pp. 515–527, SIAM, Salt Lake City, UT, USA, 2020.

[20] W. Gao, J. Zhou, Y. Lin, and J. Wei, “Compressed sensing-based privacy preserving in labeled dynamic social networks,” *IEEE Systems Journal*, vol. 6, no. 3, pp. 1–12, 2022.

[21] A. Anagnostou, I. Mollas, and G. Tsoumakas, “Hatebusters: A Web Application for Actively Reporting YouTube Hate Speech,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 5796–5798, Stockholm, Sweden, July 2018.

[22] L. Cheng, K. Shu, S. Wu, Y. N. Silva, D. L. Hall, and H. Liu, “Unsupervised cyberbullying detection via time-informed Gaussian mixture model,” 2020, https://arxiv.org/abs/2008.02642.

[23] T. Liu, K. Yu, L. Wang, X. Zhang, H. Zhou, and X. Wu, “Clickbait detection on WeChat: a deep model integrating semantic and syntactic information,” *Knowledge-Based Systems*, vol. 245, Article ID 108605, 2022.

[24] S. R. Sahoo and B. B. Gupta, “Multiple features based approach for automatic fake news detection on social networks using deep learning,” *Applied Soft Computing*, vol. 100, no. 3, Article ID 106983, 2021.

[25] S. R. Sahoo and B. B. Gupta, “Classification of spammer and nonspammer content in online social network using genetic algorithm-based feature selection,” *Enterprise Information Systems*, vol. 14, no. 5, pp. 710–736, 2020.

[26] D. Sánchez and M. Batet, “C-sanitized: a privacy model for document redaction and sanitization,” *Journal of the Association for Information Science and Technology*, vol. 67, no. 1, pp. 148–163, 2016.

[27] C. Iwendi, S. A. Moqurrab, A. Anjum, S. Khan, S. Mohan, and G. Srivastava, “N-sanitization: a semantic privacy-preserving framework for unstructured medical datasets,” *Computer Communications*, vol. 161, pp. 160–171, 2020.

[28] N. Rodprayoon, “Communication via self-disclosure behavior of micro-influencers on social media in Thailand,” *Modern Applied Science*, vol. 14, no. 2, pp. 49–56, 2020.

[29] X. Cao, Z. Luo, J. Qiu, and Y. Liu, “Does ostracism impede Chinese tourist self-disclosure on WeChat? The perspective of social anxiety and self-construal,” *Journal of Hospitality and Tourism Management*, vol. 50, pp. 178–187, 2022.

[30] H. Hong, W. Bao, Y. Hong, and Y. Kong, “Privacy attributes-aware message passing neural network for visual privacy attributes classification,” in *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, no. 2, pp. 4245–4251, Milan, Italy, January 2021.

[31] R. G. P. Livio Bioglio, “Analysis and classification of privacy-sensitive content in social media posts,” *EPJ Data Science*, vol. 11, no. 12, pp. 302–324, 2022.

[32] K. Wang, J. Wu, T. Zhu, W. Ren, and Y. Hong, “Defense against membership inference attack in graph neural networks through graph perturbation,” *International Journal of Information Security*, vol. 22, no. 2, pp. 497–509, 2023.

[33] D. Li, Q. Yang, C. Li, D. An, and Y. Shi, “Bayesian-based inference attack method and individual differential privacy-based auction mechanism for double auction market,” *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 950–968, 2023.

[34] T. H. Lin, Y. S. Lee, F. C. Chang, J. M. Chang, and P. Y. Wu, “Protecting sensitive attributes by adversarial training through class-overlapping techniques,” *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1283–1294, 2023.

[35] F. Li, H. Li, B. Niu, and J. Chen, “Privacy computing: concept, computing framework, and future development trends,” *Engineering*, vol. 5, no. 6, pp. 1179–1192, 2019.

[36] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, “Roformer: enhanced transformer with rotary position embedding,” 2021, https://arxiv.org/abs/2104.09864.

[37] F. Al Zamal, W. Liu, and D. Ruths, “Homophily and latent attribute inference: inferring latent attributes of twitter users from neighbors,” in *Proceedings of the International AAAI Conference on Web and Social Media*, pp. 387–390, Quebec, Canada, October 2012.

[38] Y. Liu, “Research on privacy measurement framework in social networks,” Master’s thesis, Xidian University, Xi’an, China, 2021.

[39] K. J. Reza, M. Z. Islam, and V. Estivill-Castro, “Privacy protection of online social network users, against attribute inference attacks, through the use of a set of exhaustive rules,” *Neural Computing & Applications*, vol. 33, no. 19, pp. 12397–12427, 2021.

[40] T. Buchanan, C. Paine, A. N. Joinson, and U. D. Reips, “Development of measures of online privacy concern and protection for use on the Internet,” *Journal of the American Society for Information Science and Technology*, vol. 58, no. 2, pp. 157–165, 2007.

[41] A. Srivastava and G. Geethakumari, “Measuring privacy leaks in online social networks,” in *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2095–2100, IEEE, Mysore, India, August 2013.

[42] R. Serfontein, H. Kruger, and L. Drevin, “Identifying information security risks in a social network using self-organising maps,” in *Proceedings of the IFIP World Conference on Information Security Education*, pp. 114–126, Springer, Berlin, Germany, June 2019.

[43] M. Humbert, B. Trubert, and K. Huguenin, “A survey on interdependent privacy,” *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–40, 2019.

[44] W. Shi, J. Hu, J. Yan, Z. Wu, and L. Lu, “A privacy measurement method using network structure entropy,” in *Proceedings of the 2017 International Conference on Networking and Network Applications (NaNA)*, pp. 147–151, IEEE, Kathmandu, Nepal, October 2017.

[45] C. Wang and K. L. Ma, “HypperSteer: hypothetical steering and data perturbation in sequence prediction with deep learning,” 2020, https://arxiv.org/abs/2011.02149.

[46] A. Masoumzadeh and J. Joshi, “Preserving structural properties in edge-perturbing anonymization techniques for social networks,” *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 877–889, 2012.

[47] T. Zhang, D. Ye, T. Zhu, T. Liao, and W. Zhou, “Evolution of cooperation in malicious social networks with differential privacy mechanisms,” *Neural Computing & Applications*, vol. 35, no. 18, pp. 12979–12994, 2020.

[48] T. Gao and F. Li, “Differential private social network publication and persistent homology preservation,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3152–3166, 2021.

[49] H. Yang, Y. Huang, Y. Yu, M. Yao, and X. Zhang, “Privacy-preserving extraction of hog features based on integer vector homomorphic encryption,” in *Proceedings of the International Conference on Information Security Practice and Experience*, pp. 102–117, Springer, Berlin, Germany, June 2017.

[50] H. Y. Tran and J. Hu, “Privacy-preserving big data analytics a comprehensive survey,” *Journal of Parallel and Distributed Computing*, vol. 134, pp. 207–218, 2019.

[51] A. Vasalou, A. J. Gill, F. Mazanderani, C. Papoutsi, and A. Joinson, “Privacy dictionary: a new resource for the automated content analysis of privacy,” *Journal of the American Society for Information Science and Technology*, vol. 62, no. 11, pp. 2095–2105, 2011.

[52] A. J. Gill, A. Vasalou, C. Papoutsi, and A. N. Joinson, “Privacy dictionary: a linguistic taxonomy of privacy for content analysis,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 3227–3236, Vancouver, Canada, May 2011.

[53] G. Xu, C. Qi, H. Yu, S. Xu, C. Zhao, and J. Yuan, “Detecting sensitive information of unstructured text using convolutional neural network,” in *Proceedings of the 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 474–479, IEEE, Guilin, China, October 2019.

[54] N. Mehdy, C. Kennington, and H. Mehrpouyan, “Privacy disclosures detection in natural-language text through linguistically-motivated artificial neural networks,” in *Proceedings of the International Conference on Security and Privacy in New Computing Environments*, pp. 152–177, Springer, Berlin, Germany, June 2019.

[55] W. Li, J. Wu, and Q. Bai, “An automated privacy information detection approach for protecting individual online social network users the Japanese Society for Artificial Intelligence,” in *Proceedings of the Annual Conference of JSAI 33rd (2019)*, pp. 305–311, Japan, June 2019.

[56] J. Wu, W. Li, Q. Bai, T. Ito, and A. Moustafa, “Privacy information classification: a hybrid approach,” 2021, https://arxiv.org/abs/2101.11574.

[57] X. Li, Y. Xin, C. Zhao, Y. Yang, and Y. Chen, “Graph convolutional networks for privacy metrics in online social networks,” *Applied Sciences*, vol. 10, no. 4, p. 1327, 2020.

[58] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-Attention with Relative Position Representations,” 2018, https://arxiv.org/abs/1803.02155.

[59] J. Su, A. Murtadha, S. Pan et al., “Global pointer: novel efficient span-based approach for named entity recognition,” 2022, https://arxiv.org/abs/2208.03054.

[60] Y. Sun, C. Cheng, Y. Zhang et al., “Circle loss: a unified perspective of pair similarity optimization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6398–6407, Seattle, WA, USA, June 2020.

[61] L. Dong, N. Yang, W. Wang et al., “Unified language model pre-training for natural language understanding and generation,” *Advances in neural information processing systems*, vol. 32, pp. 13042–13054, 2019.

[62] C. Y. Lin, “Rouge: a package for automatic evaluation of summaries,” in *Proceedings of the Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004.

[63] R. Srivastava, P. Singh, K. Rana, and V. Kumar, “A topic modeled unsupervised approach to single document extractive text summarization,” *Knowledge-Based Systems*, vol. 246, Article ID 108636, 2022.

[64] F. Souza, R. Nogueira, and R. Lotufo, “Portuguese named entity recognition using BERT-CRF,” 2019, https://arxiv.org/abs/1909.10649.

[65] H. T. Phan, N. T. Nguyen, V. C. Tran, and D. Hwang, “An approach for a decision-making support system based on measuring the user satisfaction level on twitter,” *Information Sciences*, vol. 561, pp. 243–273, 2021.

[66] K. Ren, H. Li, Y. Zeng, and Y. Zhang, “Named entity recognition with CRF based on albert: a natural language processing model,” in *Proceedings of the China Conference on Command and Control*, pp. 498–507, Springer, Singapore, August 2022.

[67] J. Liu, C. Xia, H. Yan, and W. Xu, “Innovative deep neural network modeling for fine-grained Chinese entity recognition,” *Electronics*, vol. 9, no. 6, p. 1001, 2020.

[68] S. Fan, H. Yu, X. Cai et al., “Multi-attention deep neural network fusing character and word embedding for clinical and biomedical concept extraction,” *Information Sciences*, vol. 608, pp. 778–793, 2022.