Start from generating datasets: a pipeline for sensitive information detection with BERT

TEAM 6 (*Some Group*) - **"No Regrets!"**

Li Guanzhen
A0256648N
E0966232@u.nus.edu

Wei Pengbo
A0250630U
E0945812@u.nus.edu

Rong Qixian
A0250639B
E0945821@u.nus.edu

Lyv Xueyan
A0254414L
E0957025@u.nus.edu

***Abstract*—With the vast and rapid evolution of social media, sharing thoughts or opinions on social media platforms has been much easier than ever. However, the wide adoption of social media as the major communication method not only brings convenience but also poses a potential danger that leads to serious privacy disclosure. Aiming to help users take control of their shared information and to further protect users' privacy, we developed a BERT classifier that is capable of automatically identifying posts that might contain sensitive or private content. A challenge for this task is the lack of annotated dataset, so we also proposed a pipeline to generate data samples with GPT 3.5 and annotate the unlabelled data by proposing a hybrid Named Entity Recognition Method and defining a set of rules. We not only achieved high accuracy for detecting sensitive information but also conducted experiments to indicate the feasibility of training models with datasets generated by prompt learning. We also desired to propose a solution for other research which are restricted by the lack of annotated datasets.**

***Index Terms*—sensitive information, social media, text classification, BERT, prompt learning**

I. Introduction

In this digital era, the rise of social media has imperceptibly changed the way people interact with each other, and social media has become an integral part of many people's daily lives, with users spending hours each day scrolling through feeds, sharing photos and videos, and engaging with online communities. With the advent of online platforms, people can start a communication with their friends, families and even strangers around the world in a minute with just a few clicks. From posting updates about their daily lives to sharing news stories and opinions with their online networks, it is undeniable that social media has become a ubiquitous presence in modern society.

However, the widespread use of social media also comes with potential privacy risks, as individuals may unintentionally disclose private information through their online activities. From the perspective of users, it can be difficult to maintain control over what information they share and who they can share the information with, especially when the related privacy policies and regulations are loose and not clear. At most of the time, users inadvertently disclose private or sensitive content while interacting with others on social media platforms. Sensitive information detection is one approach that can prevent users' privacy to be disclosed and this can also raise awareness among users about the importance of maintaining their own privacy.

It is not an easy task to do sensitive information detection simply by training a binary classifier for the following two reasons:

1) No specific definition about "sensitive" information and "non-sensitive" information.
2) Less or even no existing annotated dataset could be found online.

On one hand, for a sentence *"Jean has a bad headache, and she is at hospital right now."*, apparently, this sentence exposes Jean's personal privacy and her location, which could potentially lead to a security threat and should be identified as containing sensitive information. Sentences like *"The man suffered from serious headaches in the following days, reported BBC News."* or *"Listening to Trump's speech gives me a big headache."* should not be identified as sensitive information since one is from the news report and the other one does not really mean the continuous pain in the head. Whether or not the statement contains sensitive information mostly depends on the specific semantic context of the sentence. On the other hand, as this topic involves sensitive and personal information, a large majority of the existing datasets are private and not open to the public.

Nevertheless, the sensitive information detection task in the social media domain is not trivial mainly for 2 reasons: 1) there is no existing dataset exclusively for sensitive information detection in the social media domain; 2) judging if a token (or phrase) is sensitive largely depends on its contexts, i.e. maybe a token is sensitive in some contexts (e.g. Mary suffered from a headache yesterday.), while not in other scenarios (e.g. Mary has a headache on today's lecture). Fortunately, the rapidly advancing GPT models these days have proposed a possible solution to overcome the challenge.

In the case of insufficient data, some research construct new datasets via Prompt-learning GPT models recently [1] [2], which indicates a method to supplement data for this task. What's more, the method seems to match this task even better than the existing research for the following two reasons. For one thing, a general shortcoming of this method is we can't guarantee the samples GPT generates are ground truths, but the shortcoming doesn't exist in this task because we only need

seemingly true sensitive information samples, and we needn't (and had better not) use real sensitive information, to avoid privacy leak to Large Language Models (LLMs) [3]. For another thing, GPT is extraordinarily good at capturing mapping relationships within a long sentence and easily generates the seemingly real sensitive information samples we need.

Specifically, for the first challenge (lack of annotated datasets), we firstly constructed an unlabelled dataset in 2 steps: 1) extract samples from multiple existing datasets, and merge them together; 2) generate samples with sensitive information format via prompt learning. Then, we assigned each sample a label based on the rules we define. For the second challenge (depending on contexts), we aspired to supplement samples on different topics to increase the model's ability to judge sensitive information by considering the semantic contexts.

In this paper, we implemented a BERT-based text classifier that can automatically check for such sensitive information, which helps to prevent users' privacy from being exposed on social media inadvertently by alerting users to the potential privacy breaches. First, to process the dataset without labels, we proposed a hybrid NER (Named Entities Recognition) Method and extended the pre-trained models' labels by generating new samples. Then, based on the NER results and the defined rules, we annotated the data and got a labeled dataset. With constructed dataset, we eventually train a BERT-based model with multi-head attention that can map each input text to a binary label of "sensitive" or "non-sensitive". If a post is predicted as sensitive, the hybrid NER module can also highlight the NE in the sentence to remind users that which part is sensitive.

Our contributions for this task can be summarized as follows:

1) We build a hybrid NER model for preliminary data labeling.
2) We construct an annotated dataset for sensitive information detection in the social media domain.
3) We improved the labeling rules in existing research by taking a sentence's triples relationships into account.
4) We implement and fine-tune a BERT text classifier that can accurately detect sentence-level sensitive information.

II. Related work

A. Privacy Concerns on Social Media

Privacy issues of social media have been a wide concern for long. Though laws and requirements on privacy on social media have been introduced, the privacy disclosure from the user side emerge endlessly. In most cases, this is because users lack a comprehensive understanding of whom will be able to access and how such a body of information will be used. [4] The very nature of social media and the psychological characteristics of users often lead them to ignore possible privacy issues. For instance, users carelessly give away their name combined with their location, which can lead to users' identification [5].

Therefore, it is meaningful and urgent that we develop a method to point out possible sensitive contents for users and help them understand the privacy.

B. Named Entity Recognition

Named entity recognition (NER) refers to the task of identification followed by classification of various name entities from text [6]. Named entities like names, location, symptoms etc. are highly possible to disclose personal privacy or organizational secretes [4] [7]. Ji-sung Park [8] and his team has applied NER to classify words in a document into categories consisting name, location and so on, and considered some of these categories as sensitive information. This approach showed good performance identifying sensitive information in unstructured data(e.g. sentences).

Inspired by these studies, we used Bert-based NER to identify specific categories of name entities during our data pre-processing.

C. Sensitive Data Detection

For natural language data, to identify the sensitive information in the free text is quite a huge obstacle. [7] For example, in the medical domain, besides sensitive features such as names of persons and places, information about patients symptoms, diagnoses and treatments should also be recognized as privacy. The following paragraphs briefly introduced some approaches to detect sensitive information.

Nuhil Mehdy [9] and his team designed a multichannel convolutional neural network as a classifier to identify short texts that have personal, private disclosures. Taking sentence level context into consideration innovatively, they extracted features such as part-of-speech, syntactic dependencies, and entity relations to train the CNN model on a human-labeled dataset and achieved an accuracy rate of 93% accuracy. However, this model may not be as effective on short contents or in different contexts than those used in the study.

Another classifier based on the Transformer deep-learning model was developed by Michael Petrolini [10] and his team. In addition to sensitive data detection, this classifier is able to recognize the particular type of sensitive topic and enable researchers to have a better knowledge of the data. However the merging policy it uses to combine information extracted from different sources is still too simple to catch the relationship private information and its topics.

From the perspective of information theory, David Sanchez [11] and his team presented a detection method that can be applied to a general-purpose corpus, which has a wide coverage of almost any possible up-to-date term. Their main highlights are mathematically formulating what they consider sensitive information and how it can be applied to detect potentially sensitive textual entities.

Alfonso Guarino [12] and his colleagues also paid attention to data topics. They proposed a novel approach using sentence embedding techniques and containing four modules: the Keyword module, the Topic module, the Sensitiveness module and the Personalization module, which take a closer look at

the relation between topics and privacy, as well as privacy degree.

Based on these research results, we reckon that the first step of our work should be determining the definition of sensitive information, and label the collected data with 'sensitive' or 'non-sensitive' tag. After this, we will train a text classifier using labeled data for future sensitive information detection.

D. Bert-based Text Classifier

BERT (Bidirectional Encoder Representations from Transformers), which is a language representation model called BERT, was introduced by Jacob Devlin [13] and his team in 2019. By jointly conditioning on both left and right context in all layers , BERT is able to pre-train deep bidirectional representations from unlabeled text. Experiments have shown that BERT is conceptually simple and empirically powerful, and outperforms many other NLP methods.

After BERT came out, it became a popular tool to do NLP related tasks and showed satisfactory results.

Ping Huang [14]and his team combined BERT with CNN (convolution neural networks) to build a text sentiment and conducted sentiment analysis on a large movie review dataset. The BERT-CNN model achieved an accuracy as high as 86.67%.

In the area of text classification, an attempt of BERT application is made by Kuncahyo Setyo Nugroho [15] and h is team. They classified a big text of news topics with fine-tuning BERT used pre-trained models.

Another direction is content detection. One task on this topic is hate speech detection task done by Ommel Hernandez Urbano Jr. and his team [16]. Based on BERT, they proposed an automatic hate speech detection method targeted at speech transcribed from Tagalog TikTok videos and achieved an F1 scores of 61%. Another task is about cyberbullying remarks detection and Ziyang Feng [17]and his team designed a BERT-based hierarchical attention fusion network.

Since our project is related to content detection and BERT has shown good performances, we decided to apply BERT to solve our tasks.

III. Method

In this project, we proposed a pipeline (Fig. 1) for sensitive information detection. In the training phase, we formulate the task as a binary classification problem and implement BERT to detect sensitive information at the sentence level (assign a label $\in \{sensitive, insensitive\}$). However, one challenge for us is that there hasn't been an existing dataset exclusively for sensitive information detection in the social media domain yet. Hence, before solving the binary classification task, we constructed a dataset (both from Twitter text online and prompt-learning generated samples) and implemented a Hybrid NER Model (with defined rules) to label the data samples.

In the inferring (and predicting) phase (Fig. 2), to handle the contextual privacy leakage at the document level, we first substituted all Personal Pronouns with $PERSON$ Named Entities (NEs) by Coreference Resolution and then applied the sentence-level classifier to detect sensitive information.

A. Training Pipeline

1) Data Labeling-a hybrid NER method: Named Entities are essential for sensitive information detection tasks. They always refer to particular people, places, organizations (and so on) and tend to identify sensitive or confidential information. Many existing research [18] [19] [20] discover sensitive information based on Named Entities Recognition (NER) techniques in various languages. So, at the lexical level, we proposed a hybrid Named Entities Recognition method to overcome the shortcoming of existing methods.

There have already been many off-the-shelf Named Entities Taggers, and here, we applied the NER Module of Stanford CoreNLP toolkit, a combination of rule-based systems and statistical machine learning techniques. It has generally demonstrated strong performance but still exposes some shortcomings when focusing on the social media domain: 1) can't detect accurately enough; 2) fail to cover some topics where private information is likely to leak.

More specifically, for the first problem, let's cite Name Detection as an example. Stanford CoreNLP's NER module usually fails to detect people's names if the full name is not given. However, in the social media domain, users are more likely to mention peoples' real identities via their nicknames or abbreviations, which makes the Stanford toolkits' shortcomings fatal. As shown in Fig. 3, Stanford Toolkit can only detect $PERSON$ when the full name is given. Besides, the result also indicates that the toolkit failed to detect the course number, although it can solely detect numbers well, which we'll handle with Regular Expression later.

Besides, Stanford CoreNLP NER Module doesn't perform well enough on some topics (Fig. 4) which are important for this task (such as Medical, Nation Security). There are three possible reasons: 1) Stanford Toolkit doesn't cover some topics (like the Military) and returns wrong labels; 2) owing to Domain Shift, some Named Entities which can expose privacy via social media are just regarded as normal tokens from a general aspect, as Stanford CoreNLP does; 3) Stanford NLP doesn't consider labels sequence's order enough and largely depends on the corpus (may be out-of-date), hence can't detect some newly appeared Named Entities (like Covid-19).

To alleviate the problems mentioned above, we implemented a BERT-based NER model (Fig. 5) as a supplement. We expect the BERT-based NER model to function as follows: 1) on some common labels (especially $PERSON$), outperform the Stanford CoreNLP, and supplement the undetected elements; 2) detect the Named Entities in the topics that Stanford CoreNLP doesn't cover, based on the supplemented data samples generated by GPT 3.5 (text-davinci-003).

Additionally, there are also two modules in this hybrid NER method: 1) for detecting the entities with relatively constant formats, Regular Expression is more effective than Neural-Network-based methods; (Stanford CoreNLP can't detect a mixture of numbers and letters.) 2) for **label mapping**, we defined rules (Table A) to map the results from two models (Stanford toolkit and BERT-NER) to the same set of labels.

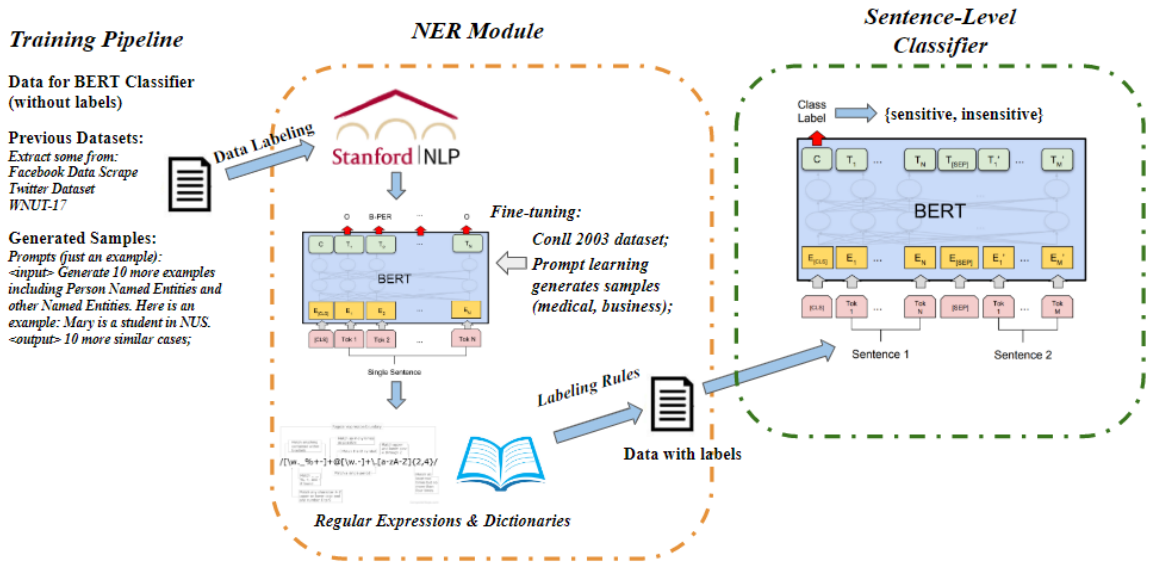

Fig. 1. The Training pipeline of sensitive information detection task

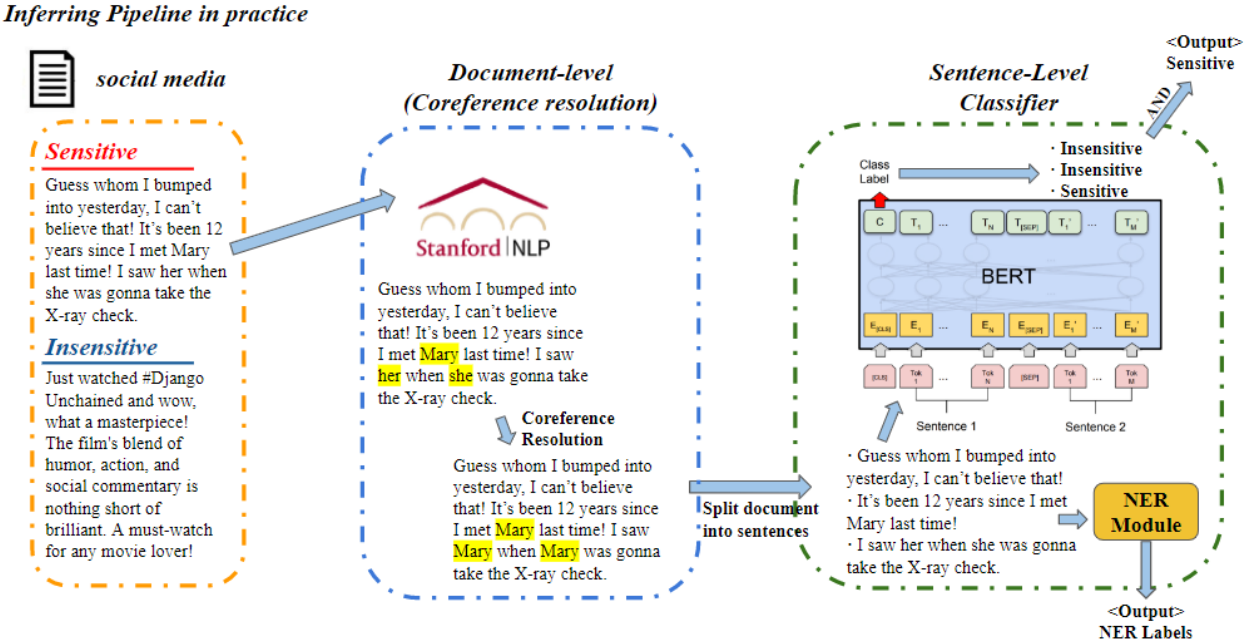

Fig. 2. The Inferring pipeline of sensitive information detection task

3) a dictionary of words that are not Named Entities but can indicate sensitive information (like hospital, bank, etc.)

2) Data Labeling-Relation Extraction: Although the NER model may successfully extract all relevant sensitive entities, it can still fail since a sentence can have different meanings even with the same set of entities. For instance, consider the sentences "I don't like San Francisco" and "I have been living in San Francisco for a long time." Both sentences include a person entity and a location entity, but the first sentence is a non-disclosure sentence, whereas the second one is a disclosure sentence. It is difficult to distinguish the difference between the two examples by considering only entity-level information. Thus, we apply relation extraction to consider sentence-level information.

Relation extraction from sentences plays a vital role in semantic analysis, which aims to identify and extract the relationships between entities mentioned in text. Common relation extraction technologies, including rule based, machine learning based, and deep learning based, have achieved remarkable performance in extracting specific type of relations.

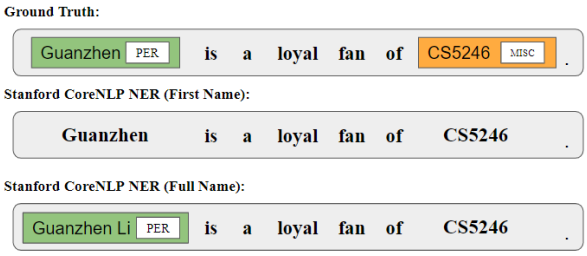

Fig. 3. Stanford CoreNLP's performance on People's Names Detection. It can only detect PERSON when full names are given.

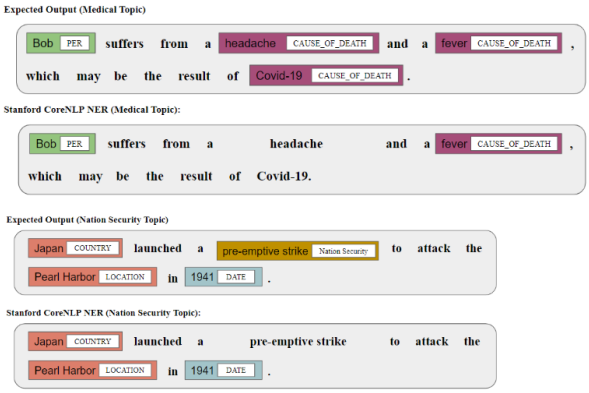

Fig. 4. Stanford CoreNLP NER misses some Named Entities on certain topics

However, these technologies still under-performed on open-relation extraction task.

Prompt learning. Large-size deep learning language models , e.g. ChatGPT, have shown unbelievable ability on solving document analysis task. By leveraging the advantages of large-size models, promoting learning using prompts or input templates to guide the training and inference process instead of training a model from scratch. In this project, we predefined the input templates to guide the model towards extracting open-relation from sentences. Instead of zero-shot learning, we adopt few-shot learning to help the language model better understand the collected data and this specific task. Specifically, we created some labeled relation extraction examples with corresponding named entities to generalize from these few examples and recognize new relations with similar features. Here is an example of the created data with the input template.

After putting the created labeled data, the large-size language model should have the ability to predict unknown relations from given sentences. We record the relation predictions and further utilize them to assign sensitive labels for short-text in the collected dataset.

3) Data Labelling-Rules: After obtaining the extracted relations from each short-text, we adopt rule-based algorithm to assign sensitive labels for each data. In contrast to prior research [21], our approach not only employs the sensitive-related entities but also considers the relations present within a given short text. To better understand the rules, we define the following definition:

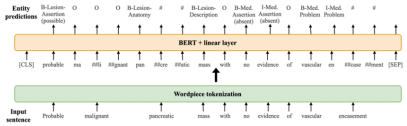

Fig. 5. The architecture of BERT-NER model

Input: *John is from New York City.*
<John, Person> <New York, Location>
Output: *<Person, from, Location>*

Fig. 6. An Example of the created data with the input template.

Definition 1. (Disclosure Related Entity Type) - *Each $DRET_f$ is a set of DREs that belong to a type f,, where $f \in F$ = Person, Organization, Location, Medical, Date, Time, Financial Information. Having D as an infinite set of all possible DREs then*

$$\forall DRE_d \in D \nexists i, j \in F \; where \; i \neq j, DRE_d \in DRET_i \cap DRET_j$$

Definition 2. (Sensitive) - *Let R be the extracted relation from a short-text S and R = (e1, r, e2). S contains sensitive information if it satisfies the following condition:*

$$\exists \, e1, e2, where \; e1 \in DRET_{PERSON}, \wedge e2 \in \cup_{f \in F} DREF_f$$

or

$$\exists \, e1, e2, where \; e1 \in DRET_{ORG}, \wedge e2 \in DRET_{Fin.Info}$$

To label a short-text as sensitive (Definition 2), We check the extracted relation tuples from a short-text. If one of the extracted relation tuple contains a Person entity and any other entities or a Organization entity and a Financial information entity, we label this short-text as sensitive. A more structured guideline for manual labeling is given below:

1) Start with an example short-text. Check if the extracted relations contains one or more DRE which falls into the set of DRET (Definition 1).
2) If in a relation tuple, one of the DREs is type of Person or one of the DREs is type of Organization with a Financial information type DRE, then go to step 3, otherwise label it as non-sensitive
3) Label the short-text as sensitive and return to step 1.

4) Sentence-Level Classifier: After assigning a sensitive label to each short text, we utilize a sentence-level classifier to learn sensitive patterns from the labeled data. By incorporating the labels derived from the fixed rule-based algorithm, the trained sentence-level classifier can improve generalization ability by learning a soft decision boundary. Moreover, in terms of complexity, the sentence level classifier can save many steps (e.g. NER, relation extraction). It can automatically detect whether a short-text sentence is sensitive or not.

Neural Networks have proven to be highly effective in handling natural language processing tasks. In particular, BERT, a state-of-the-art NLP model, has outperformed previous models like Logistic Regression and Recurrent Neural Networks.

Consequently, in this project, we employ the pre-trained BERT model and fine-tune it on the gathered dataset to determine if a short text is sensitive or not.

BERT. When compared to traditional machine learning methods like logistic regression, BERT offers a distinct advantage in that it considers not only the lexical-level information of words but also analyzes the temporal relationships between them using the attention mechanism. Furthermore, in contrast to using a recurrent network backbone, BERT avoids the issues of gradient vanishing that can occur when the input sequence length is large. This is achieved through its use of a transformer architecture, which allows for parallel computation during the training process. By leveraging these advantages, BERT outperformed most of the previous models in natural language processing tasks.

Therefore, for our sensitive information classification task, we utilize the Transformer Encoder in BERT to extract feature from text inputs, followed by passing it through a multi-layer perceptron (MLP) to determine whether a short-text contains sensitive information or not. By adopting the advanced transformer encoder, the model can capture the semantic meaning of the text inputs. The MLP layer further help learn complex decision boundaries and boost the generalization ability.

B. Inferring (Predicting) Pipeline

1) Coreference Resolution: Restricted by the BERT Classifier training dataset's average text length, it can detect sensitive information at the sentence level. Although on social media platforms, the posts tend to be short, we should also consider the contextual privacy leakage at the document level [3], in case there are some long posts.

Specifically, the following example consists of three sentences. Solely observe the three sentences separately, they don't leak sensitive information based on the rules we defined in III-A3. In the third sentence, the Personal Pronoun She actually refers to the $PERSON$ NE $Mary$ in the second sentence. Hence, it leads to contextual privacy leakage.

```
Guess whom I bumped into yesterday, I can't
believe that! It's been a long time since I
met Mary last time! I saw her when she was
gonna take the X-ray check.
```

To handle the problem, we proposed a solution for inferring phase: apply Stanford CoreNLP $coref$ module to substitute all Personal Pronouns to the corresponding $PERSON$ NEs. In this way, we expect to improve the sentence-level classifier's performance at the document level. The processed text is as follows, where the third sentence can be judged as sensitive now:

```
Guess whom I bumped into yesterday, I can't
believe that! It's been a long time since
I met Mary last time! I saw Mary when Mary
was gonna take the X-ray check.
```

IV. Data

A challenge for sensitive information detection tasks in the social media domain is lacking annotated datasets. Hence, the dataset construction and data augmentation step is a novelty in our project. Inspired by the recent surge of emerging research [2] [1] [22] which apply Prompt-Learning to generate datasets, we made an insightful attempt that if GPT (Generative Pre-trained Transformer) can help to overcome the challenge. As mentioned in section I, we came up with the idea mainly for two reasons: 1) GPT is good at generating new samples when given a certain format; 2) We don't need to guarantee the data samples are true.

A. Hybrid NER Method: BERT Fine-tuning

Data collection & Data augmentation Our BERT-based NER Neural Network is mainly trained on CoNLL-2003. CoNLL is commonly used for a collection of NLP tasks, including Part-Of-Speech labeling, NER, Dependency Parsing, etc. The samples contain four features: text, Part-Of-Speech labels, Dependency Parsing Labels, and Named Entities Labels.

For the NER task, the CoNLL only contains four labels $PER, ORG, LOC, MISC$. It covers the most common ones but doesn't consider some risky topics in the social media domain. Hence, we applied prompt learning to generate Medical and Financial labels in the CoNLL format as follows. Totally, the final dataset contains 331526 tokens (224348 for the training set, 51283 for the test set, and 55895 for the validation set).

Data Preprocessing. With data collected from different resources, we first double-checked the dataset using Stanford CoreNLP toolkit to complement some missing entity types in the original dataset. (e.g. The generated sample: Mary takes Ibuprofen produced by Charlie Industry on a daily basis. GPT only labels Mary and Ibuprofen as PER and Med separately but doesn't label Charlie Industry as ORG. By using Stanford CoreNLP for double-check, we can detect the ORG NE.)

```
A generated Medical Example:
The DT B-NP O
hospital NN I-NP O
prescribed VBD B-VP O
Azithromycin NNP B-NP B-Med
for IN B-PP O
her PRP$ B-NP O
throat NN B-NP B-Med
infection NN I-NP I-Med
. . O O
```

So, finally, the trained BERT based NER model contains the following labels: {PER, LOC, ORG, MISC, Med, Fin, DATE}

According to the distribution of the labels (Fig. 7), the labels are imbalanced, and the number of labels O is significantly larger than the others. So, the distribution points out two requirements for our experiments: 1) Be cautious about the overfitting problem; 2) Focus on the Recall, Precision, and F1 instead of Accuracy for evaluation.

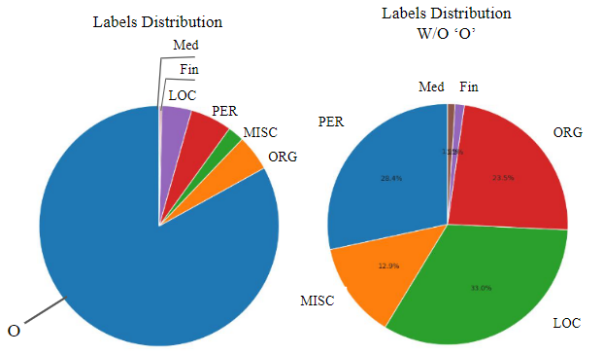

Fig. 7. Labels Distribution: the left one is labels distribution of all labels; the right one is the labels distribution without label O

B. Sensitive information classification Dataset

Data Collection. Due to the limited availability of text data containing sensitive information, we leveraged OpenAI's ChatGPT to generate 403 samples (due to limited OpenAI free trail balance) containing fake personal or company information, such as occupation, email, medical info, financial info, and others. Additionally, we randomly selected 615 tweets, and Facebook posts from an open dataset that are considered non-sensitive short-text. The overall statistic of this collected dataset is shown in Table I.

Data Preprocessing. Apart from the data generated via prompt-learning, the majority of the training data is the raw tweets collected from twitter, which is highly unstructured and contains a lot of redundant and unrelated information. With more unstructed and redundant data feeding into our classifier, it has been proven that the training process will be more unstable and the final model would perform worse. Therefore, it is crucial to pre-process the collected data to overcome the above issues. The very first step is to perform basic text cleaning on the input data. Unstructed information such as hashtag, URLs, emojis and smileys, need to be removed first since the elements mentioned above are very common on almost every social media site. Then, we do further preprocessing on the clean input data using the NLP toolkit NLTK(Natural Language Toolkit) to tokenize the sentences in a customized way that can keep important punctuations and remove redundant punctuations. Different from common text preprocessing steps, we take all the valid sequential tokens into account, which turns out to be helpful for model in learning important entity relationships and the importance of the token sequence.

TABLE I
THE COLLECTED DATASET STATISTIC.

	Sensitive samples	Non-sensitive samples	Total samples
The collected dataset	403	615	1018

V. EXPERIMENTS

A. The Hybrid NER Model

In this section, we implemented a hybrid method, i.e. we detected named entities by off-the-shelf Stanford CoreNLP NER Module and a BERT-NER model trained by ourselves at the same time. Besides, we also applied Regular Expression to detect some sensitive named entities with relatively constant formats (such as phone numbers, email addresses, bank accounts, etc.) as a supplement (No too many experiments here, so we'll illustrate that in Appendix). For the Stanford Toolkit module, we applied the 4.5.2 version (Off-the-shelf toolkit, not much to introduce here). Our experiments mainly focus on the Neural Network Module in this section.

Configurations. For training Bert-based NER models, we set the number of epochs as 16, the learning rate as 0.0001, and the training batch size as 32 (limited by available computing resources). Besides, we also tried different BERT variations' performances on this task. Besides, we set 500 steps warmup (make the learning rate tiny at first, and increase gradually later), to get more stable training parameters.

Baseline. We adopt various baselines.

- Robustly Optimized BERT Pretraining Approach (RoBERTa)
- Cross-lingual Language Model RoBERTa (XML-RoBERTa)
- Bidirectional and Auto-Regressive Transformers (BART)
- Span-based BERT (spanBERT)

Evaluation. According to the EDA part, the dataset for NER is extremely imbalanced (Fig. 7) and the label O is significantly more than the other labels (normal for NER task). Hence, commonly used Accuracy is not a suitable metric here, because we care more about the minority labels. So, we choose to evaluate our hybrid NER method by Recall, Precision, and F1 Score (Micro and Macro).Besides, we exclusively extracted the metrics for the $PERSON$ label and listed them in the results.

Let L denote the set of labels, TP_i denote the number of samples correctly predicted as positive for label i, FP_i denote the number of samples that are incorrectly classified as positive for label i, FN_i denote the number of samples that are incorrectly classified as negative for label i. So the metrics can be calculated as:

$$Recall_{PER} = \frac{TP_{PER}}{TP_{PER} + FN_{PER}}$$

$$Precision_{PER} = \frac{TP_{PER}}{TP_{PER} + FP_{PER}}$$

$$F1_{PER} = 2 * \frac{Precision_{PER} * Recall_{PER}}{Precision_{PER} * Recall_{PER}}$$

$$Avg - MacroF1 = \frac{1}{|L|} \sum_{i \in L} F1_i$$

$$Avg - MicroF1 = \frac{2 \times \sum_{i \in L} TP_i}{2 \times \sum_{i \in L} TP_i + \sum_{i \in L} FP_i + \sum_{i \in L} FN_i}$$

Results. Table III reports different NER models' performance. Generally, BERT is a better option for this task. According to the rules we have defined when labeling data, $PERSON$ is crucial, and thus we exclusively list the metrics on $PERSON$ here. An interesting observation here is BERT outperforms its variations (which are proposed for improving BERT), and the possible reasons are as follows: 1) BERT is proposed earlier than the other models and has been extensively studied and sufficiently fine-tuned; 2) BERT is trained on

a larger corpus than the others; 3) BERT has a more straightforward architecture than some models (BART), which makes the fine-tuning more efficient; 4) NER just considers different tokens in a sentence and doesn't need to consider the relationships between sentences, so BART incorporates a sentence-level corruption objective, which leads to a worse performance than BERT.

Table IV shows BERT's performance on different labels. Although the model's performance on the labels we supplemented (Med & Fin) is still a little worse than the original ones (PER, LOC, ORG, MISC), its metrics scores are still relatively high considering the supplemented data's tiny proportion (Fig. 7), which indicates our proposed method's feasibility to extend existing models' labels by supplementing training data.

Error Analysis. Generally, I observed three types of errors (examples as follows): 1) wrong boundary; 2) can't detect some OOV tokens when meeting them for the first time; 3) don't consider the order of labels. More specifically, in Example 1, the model predicted the 'prescribed' and for' as 'B-Med' as well. We conjectured that it may be because 'prescribed' has a high frequency to be followed by a 'B-Med' in the training set. Besides, Example 1 also exposes an error in that B-Med is followed by another B-Med. In Example 2, the model also fails to detect some 'rare tokens', especially the ones that just appear in the test set, but do not appear in the training set. For the exposed error 2) and 3), a reason may be that the model doesn't consider the order of labels enough. (A solution may be adding a CRF layer, but it's hard to converge in our implementations, which may be limited by the dataset.)

```
Example 1:
1) Wrong boundary & 3) wrong labels order:
The hospital prescribed Azithromycin for her
throat infection.
Output: O O B-Med B-Med B-Med O B-Med I-Med .
Example 2:
2) First time met OOV
The nurse recommended taking Amoxicillin
for the patient.
Output: O O O O O O O O .
```

B. Sensitive information classifier

In the previous work, BERT has achieved remarkable performance on natural language processing, especially text classification task.Instead of training the Bert based classifier from scratch, we downloaded the pre-trained weights and fine-tuned it on the collected dataset (in Section IV-B).

Configurations. For training the Bert classifier, we set the number of epochs as 16, the learning rate as 0.0001 (for fine-tuning purpose), and the training batch size as 32 (limited by available computing resources).

Baselines. We adopt various baselines for the text classification task, including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Long Short-Term Memory Network (LSTM). The default word embeddings in the baseline models are from glove6B.

Evaluation. For this classification task, we utilized the metrics Recall, Precision, F1 score as mentioned in Section V-A because the collected data also is imbalanced.

Results. Table IV reports the sensitive detection performance of the BERT based classifier and other baseline models. From the table, BERT based classifier consistently outperforms various baselines for sensitive information classification on the collected dataset. Specifically, BERT based classifier achieves 7% $\sim$ 9% improvements w.r.t Precision, Recall and F1-score metrics compared to the baselines on the collected dataset.

Error Analysis After evaluation, we identified several cases in which the trained model was unable to learn. First, misclassify non-sensitive text as sensitive, if the semantic meaning of the text is complex. For example, "Listening to Trump's speech gives me a big headache." The classifier still cannot identify whether this person really has headache or not. The reason behind this is that the rule-based labeling algorithm wrongly assign this sentence as "sensitive" as the relation tuple contains "Person" and "Medical". Thus, we need to considering more information (e.g. sentence structure) and enrich the collected dataset to overcome this issue. Second, due to limited sensitive-related NE detected, the classifier failed to detect sensitive information when there are some sensitive NEs the model doesn't meet before. For example, "I got unexpected divorced after 2 years of relationship". In our framework, we didn't consider the personal relationship disclosure as sensitive information during the NER and rule-based labeling algorithm. We need to further detect more sensitive related NEs categories and collect more different sensitive sentences to solve this issue.

Important hyper parameters tuning. We also conducted an evaluation of two key hyperparameters in the Bert-based classifier, namely the input sequence length in Bert and the number of hidden neurons in the MLP layer. First, the optimal input sequence length may vary depending on the dataset, as a small input sequence may fail to capture all the relations between the words, while a large input sequence may result in overfitting. Second, the number of hidden neurons controls the model's capacity and the complexity of the learned function. Thus, we tested the model's performance with adjusting the two hyper parameters on cross-validation 5 setting.

For testing the input sequence length hyper parameter, we set the input length from 4 to 32 because the average length in the collected dataset is 24.73. The results are depicted in Figure 8. As can be observed, the best performance is achieved when the input sequence length is 32, which is logical given the average length of texts in our dataset.

To test the impact of the number of hidden neurons in the MLP layer, we experimented with a range of hidden neuron values from 32 to 256. The results are shown in Figure 9. We observed that the classifier's performance remained stable after the number of hidden neurons reached 64. The reason is that the collected dataset may too simple for this Bert based classifier and there no need to add more neurons to learn complex functions.

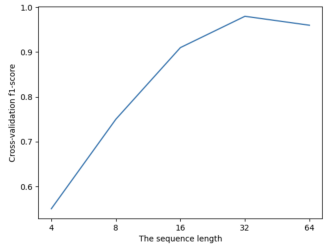

Fig. 8. Input sequence length parameter v.s. F1-score

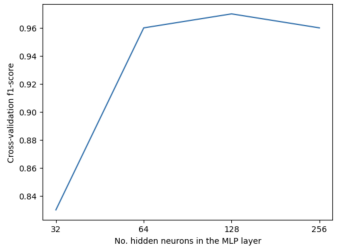

Fig. 9. No. hidden neurons v.s. F1-score

VI. Discussion

Q1: Does the sensitive information classifier can really detect sensitive information from text?

TABLE II
PERFORMANCE OF DIFFERENT BERT-BASED NER MODELS. ALL THE METRICS FOR *PERSON* HERE IS THE AVERAGE VALUE OF *B-* AND *I-*.

	Average Micro F1	Average Macro F1	PERSON F1	PERSON Precision	PERSON Recall
BERT	**0.914946**	**0.879553**	**0.974301**	**0.976284**	**0.972344**
RoBERTa	0.907475	0.872727	0.967555	0.962366	0.972808
BART	0.903966	0.873200	0.959904	0.948995	0.971069
SpanBERT	0.886861	0.845057	0.956287	0.958311	0.954294
XLM-RoBERTa	0.908049	0.878370	0.964214	0.960511	0.967969

TABLE III
BERT-NER'S PERFORMANCE ON DIFFERENT LABELS

B- or I-	PER		ORG		LOC		MISC		Med		Fin	
	Precision	recall	precision	recall	Precision	recall	precision	recall	Precision	recall	precision	recall
B-	0.977914	0.955635	0.896127	0.892461	0.929082	0.935213	0.804318	0.849003	0.907407	0.816667	0.934426	0.780822
I-	0.987900	0.989610	0.873357	0.872315	0.819149	0.902344	0.597786	0.750000	0.800000	0.235294	0.800000	0.923077

TABLE IV
SENSITIVE INFORMATION CLASSIFICATION ON THE COLLECTED DATASET.

	Precision	Recall	F1-score
MLP	0.834951	0.831837	0.830982
CNN	0.907395	0.912698	0.907209
LSTM	0.898203	0.890723	0.893293
BERT	**0.970292**	**0.971284**	**0.970257**

For solving this question, we conduct an experiment to visualize the attention weights with respect to each word in the input sentence. In this experiment, the darker colors indicates the larger attention weights. From Figure 10 to Figure 12, the three instances demonstrate the ability of the BERT-based classifier to detect sensitive terms in a sentence. The classifier accurately identified health-related words such as "sore" and "tongue" in the first example, address-related terms like "239 38 main street" in the second example, and occupation-related words such as "artist" and "paintings" in the third example. Furthermore, the model was able to recognize Person related terms, such as "my aunt" and "Maria," from the examples.

Fig. 10. Visualization attention weights for example 1

Fig. 11. Visualization attention weights for example 2

Fig. 12. Visualization attention weights for example 3

Q2: How do our Hybrid NER method improve the off-the-shelf toolkit?

Metrics NER Models achieve an extremely high Precision and Recall for label $PERSON$, and also perform well on the labels we defined ourselves. Hence, the Neural Network can function as we expected in two ways: improving Stanford CoreNLP's accuracy and extending NER models to the uncovered labels which are important for this task.

Cases As mentioned above, Stanford CoreNLP is not good at detecting people's names when the full name is not given. Our trained Neural Network can help to solve the problem. An example is as follows (Fig. 13). For the case: *Guanzhen is a loyal fan of CS5246.* The Stanford CoreNLP can't detect the PER *Guanzhen*, or the string *CS5246*. With our proposed hybrid NER method, the Neural Network can help to detect PER, and the Regular Expression can detect $CS5246$.

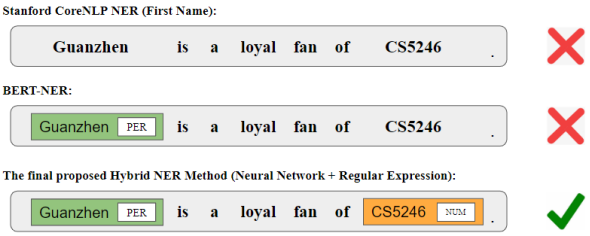

Fig. 13. Hybrid Method helps to improve Stanford CoreNLP's name detection

Besides, Stanford CoreNLP can't detect two NE labels (Medical and Financial), which are very likely to leak personal sensitive information on social media platforms. With fine-tuned BERT-NER, the hybrid method covers the two topics now. The followings are two examples (Fig. 14).

Q3: Is it feasible to extend a pre-trained NER model by Prompt-learning Data Augmentation?

Samples generation Prompt learning is good at generating samples in the same format as the given examples. During the experiments, we got the following interesting but insightful observations: 1) zero-shot (without specific examples in the prompt) is capable enough to generate samples, but can't guarantee the samples in the pre-defined format as we expected; 2) state the requirement for the samples explicitly in the prompt can instruct LLMs generate more qualified samples (the ones that contain necessary elements); 3) Compared to one-shot, few-shot yields more diverse generated

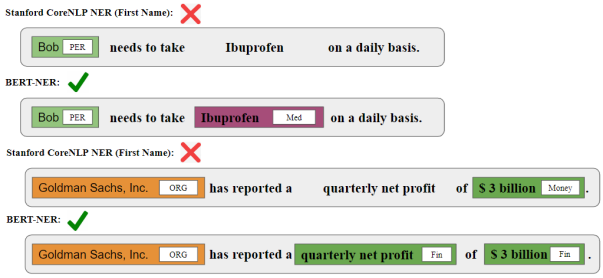

Fig. 14. Hybrid Method can detect Medical NEs and Financial NEs

samples, as one-shot can occasionally confuse models regarding which element to substitute.

Models' performance For the two labels we supplemented (Med and Fin), the fine-tuned model also achieves a relatively high Precision and Recall, which indicates the feasibility to extend pre-trained Large Language Models by supplementing data. However, it's noticeable that the Recall for label $I-Med$ is significantly lower than the others (just 0.23) and the model's overall performance on Medical Labels is worse than that for Financial Labels. There are two possible reasons: 1) Medical terms tend to be single words, and there are few I-Med tokens in the dataset; (Medical NE: Ibuprofen; Financial NE: $3 billion) 2) most financial NEs are combined with common tokens (share portfolio) while most Medical NEs are rare tokens and even OOV (out of vocabulary).

Q4: Does the GPT model can really extract open relations from text with few shot learning?

Here we provided a few cases that shown the open relation extraction ability of the Chat-GPT from short text with few shot learning.

From Figure 15, when we gave some sentences related to person's activities, the Chat-GPT can accurately extract this kind of relation from a unseen sentence, which is quite amazing. Specifically, when input this sentence "Rachel visited her friend Sarah in London. <Rachel, Person> <Sarah, Location> <London, Location>", it outputs "<Person, visit, Person, in, Location>", which is a more complicated relation. From Figure 16, when we gave some sentences related personal ID or mobile phone, we tested it with a sentence doesn't contain this relation. The result from Chat-GPT is quite correct because the mentioned email address is not associated with the Person even if we provided the ID and Person NEs. The two examples provided show that Chat-GPT is capable of extracting open relations from text, which supports the validity of our decision to use GPT for open relation extraction.

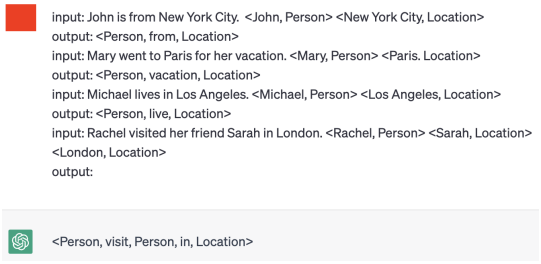

Fig. 15. The result of Chat-GPT relation extraction in case 1

VII. Conclusion & Limitation

In this report, we proposed a valid pipeline for labeling data and training a classifier for sensitive information detection in the social media domain. An important as well as insightful conclusion of this project is: GPT (Prompt-learning) is an almost perfect tool for data augmentation, and even generating the entire dataset. We conducted

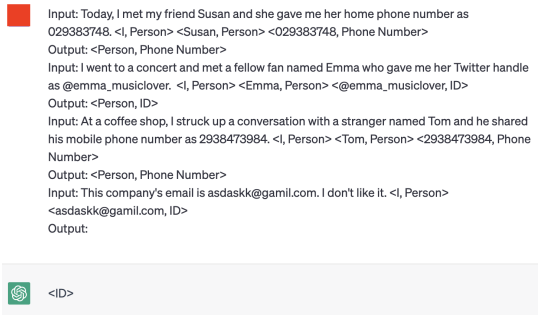

Fig. 16. The result of Chat-GPT relation extraction in case 2

<INPUT> Help me to remove the tokens between <> labels and rewrite the sentence to make it fluent: <PERSON> suffers from <Med> and needs to take <Med> on a daily basis.
<OUTPUT> The person is afflicted with a medical condition and requires a daily medication regimen.

<INPUT> Help me to remove the tokens between <> labels and rewrite the sentence to make it fluent: <PERSON> bought <Fin> at <ORG> yesterday.
<OUTPUT> A person made a purchase at an organization yesterday.

Fig. 17. Rewriting with GPT

experiments to indicate that it's indeed feasible to train and extend classification LLMs and NER LLMs by generated samples. We hope to provide inspiration for research on tasks that are currently restricted by the lack of annotated datasets. Besides, we also evaluated and demonstrated the ability of the BERT based classifier on detecting sensitive information from text by comparing with traditional deep learning models and visualizing the attention weight w.r.t each word in the input sentence.

However, there are still some limitations for us to improve in the future: 1) In the data labeling step, we tried to detect all Named Entities by using multiple methods simultaneously, which can lead to a high Recall. This means it's likely to judge a piece of insensitive information as sensitive in practice (a normal problem for outlier detection tasks). 2) For some steps, we are limited by the imperfection of the off-the-shelf toolkits themselves (like the Stanford CoreNLP $coref$ Module). 3) We have just supplemented NEs under two topics (Medical and Financial) for now, which is absolutely not complete yet. But, we have justified that the method is workable and have proposed a solution for extending the current NER models. 4) Compared to real-world text samples, the text samples generated by GPT may be too simplistic and may not cover all possible scenarios. When working with real-world datasets, our classifier may fail to detect sensitive information. To address this, we plan to apply a multi-channel network that considers auxiliary information, such as sentence structure, to better understand the input text.

There are several avenues for future research that could build upon the findings of this study. Firstly, how to generate datasets or conduct data augmentation from the aspect of prompt engineering or prompt generation may be a direction in the future (The samples generated by hand-crafted prompts are not diversified enough). Besides, a more systematic summary of the Named Entities that are likely to leak privacy in the social media domain can empower the models to cover more NEs and improve the current constructed dataset. In addition, we plan to investigate the use of a multi-channel network for handling more complex text inputs in real-world scenarios. To move a step further, we also noticed GPT's strong potential for sensitive information rewriting (Fig. 17), which may be a direction in future.

REFERENCES

[1] Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*, 2021.

[2] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *arXiv preprint arXiv:2202.04538*, 2022.

[3] Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. What does it mean for a language model to preserve privacy? In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2280–2292, 2022.

[4] Giovanni Livraga, Alessandro Motta, and Marco Viviani. Assessing user privacy on social media: The twitter case study. In *Open Challenges in Online Social Networks*, OASIS '22, page 1–9, New York, NY, USA, 2022. Association for Computing Machinery.

[5] Katerina Vgena, Angeliki Kitsiou, Christos Kalloniatis, and Dimitris Kavroudakis. Disclosing social and location attributes on social media: The impact on users' privacy. In Sokratis Katsikas, Costas Lambrinoudakis, Nora Cuppens, John Mylopoulos, Christos Kalloniatis, Weizhi Meng, Steven Furnell, Frank Pallas, Jörg Pohle, M. Angela Sasse, Habtamu Abie, Silvio Ranise, Luca Verderame, Enrico Cambiaso, Jorge Maestre Vidal, and Marco Antonio Sotelo Monge, editors, *Computer Security. ESORICS 2021 International Workshops*, pages 138–157, Cham, 2022. Springer International Publishing.

[6] Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. Named entity recognition and relation extraction: State-of-the-art. *ACM Comput. Surv.*, 54(1), feb 2021.

[7] Bianca Buff, Joschka Kersting, and Michaela Geierhos. Detection of privacy disclosure in the medical domain: A survey. In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2020)*, pages 630–637. SCITEPRESS, 2020.

[8] Ji-sung Park, Gun-woo Kim, and Dong-ho Lee. Sensitive data identification in structured data through genner model based on text generation and ner. In *Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things*, CNIOT2020, page 36–40, New York, NY, USA, 2020. Association for Computing Machinery.

[9] Nuhil Mehdy, Casey Kennington, and Hoda Mehrpouyan. Privacy disclosures detection in natural-language text through linguistically-motivated artificial neural networks. In Jin Li, Zheli Liu, and Hao Peng, editors, *Security and Privacy in New Computing Environments*, pages 152–177, Cham, 2019. Springer International Publishing.

[10] Michael Petrolini, Stefano Cagnoni, and Monica Mordonini. Automatic detection of sensitive data using transformer- based classifiers. *Future Internet*, 14(8), 2022.

[11] David Sánchez, Montserrat Batet, and Alexandre Viejo. Detecting sensitive information from textual documents: An information-theoretic approach. In Vicenç Torra, Yasuo Narukawa, Beatriz López, and Mateu Villaret, editors, *Modeling Decisions for Artificial Intelligence*, pages 173–184, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[12] Alfonso Guarino, Delfina Malandrino, and Rocco Zaccagnino. An automatic mechanism to provide privacy awareness and control over unwittingly dissemination of online private information. *Computer Networks*, 202:108614, 2022.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[14] Ping Huang, Huijuan Zhu, Lei Zheng, and Ying Wang. Text sentiment analysis based on bert and convolutional neural networks. In *2021 5th International Conference on Natural Language Processing and Information Retrieval (NLPIR)*, NLPIR 2021, page 1–7, New York, NY, USA, 2022. Association for Computing Machinery.

[15] Kuncahyo Setyo Nugroho, Anantha Yullian Sukmadewa, and Novanto Yudistira. Large-scale news classification using bert language model: Spark nlp approach. In *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*, SIET '21, page 240–246, New York, NY, USA, 2021. Association for Computing Machinery.

[16] Rommel Hernandez Urbano Jr., Jeffrey Uy Ajero, Angelic Legaspi Angeles, Maria Nikki Hacar Quintos, Joseph Marvin Regalado Imperial, and Ramon Llabanes Rodriguez. A bert-based hate speech classifier from transcribed online short-form videos. In *2021 5th International Conference on E-Society, E-Education and E-Technology*, ICSET 2021, page 186–192, New York, NY, USA, 2021. Association for Computing Machinery.

[17] Ziyang Feng, Jintao Su, and Junkuo Cao. Bhf: Bert-based hierarchical attention fusion network for cyberbullying remarks detection. MLNLP '22, page 1–7, New York, NY, USA, 2023. Association for Computing Machinery.

[18] Lelio Campanile, Maria Stella de Biase, Stefano Marrone, Fiammetta Marulli, Mariapia Raimondo, and Laura Verde. Sensitive information detection adopting named entity recognition: A proposed methodology. In *Computational Science and Its Applications–ICCSA 2022 Workshops: Malaga, Spain, July 4–7, 2022, Proceedings, Part IV*, pages 377–388. Springer, 2022.

[19] Mariana Dias, João Boné, João C Ferreira, Ricardo Ribeiro, and Rui Maia. Named entity recognition for sensitive data discovery in portuguese. *Applied Sciences*, 10(7):2303, 2020.

[20] Sudeshna Das and Jiaul H Paik. Context-sensitive gender inference of named entities in text. *Information Processing & Management*, 58(1):102423, 2021.

[21] Nuhil Mehdy, Casey Kennington, and Hoda Mehrpouyan. Privacy disclosures detection in natural-language text through linguistically-motivated artificial neural networks. In *Security and Privacy in New Computing Environments: Second EAI International Conference, SPNCE 2019, Tianjin, China, April 13–14, 2019, Proceedings 2*, pages 152–177. Springer, 2019.

[22] Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*, 2022.

APPENDIX

TABLE V
LABEL MAPPING FOR DIFFERENT MODELS IN HYBRID NER METHOD

Labels	Stanford CoreNLP NER	BERT-NER	Regular Expression
PERSON	PERSON	PER	-
ORGANIZATION	ORGANIZATION	ORG	-
LOCATION	LOCATION	LOC	-
Med	CAUSE_OF_DEATH	Med	-
Fin	MONEY	Fin	-
MISC	DATE, TIME, PERCENT, MISC	MISC	NUM, Phone, EMAIL, ADDRESS, etc.

```
Define Regular Expressions
NUM+Letter      ^[a-zA-Z]+\d+$|^\d+[a-zA-Z]+$
Phone Number     ^\(\d{3}\)\s?\d{3}-\d{4}$|^\d{3}-\d{3}-\d{4}$
                 ^\(\d{3}\)\s?\d{3}-\d{4}$|^\d{3}-\d{3}-\d{4}$
                 ^(\(?(?:0(?:0|11)\)?[\s-]?\(?|\+)(44)\)?[\s-]?
                 )?\(?0?\d{4}\)?[\s-]?\d{3}[\s-]?\d{4}$
                 ^(\d{4}-|\d{3}-)?(\d{8}|\d{7})$
EMAIL           ^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$
POSTCODE        ^\d{5}(-\d{4})?$
                 ^[A-Z]{1,2}\d[A-Z\d]? \d[A-Z]{2}$
                 ^[A-Z]\d[A-Z] \d[A-Z]\d$
EMOJI           ^/[:;]+-?[\)DPp]/$
                 ^[:;]-?[\)\(DPOo3\|]$
Hashtag         ^\B#\w*[a-zA-Z]+\w*$
```

TABLE VI
CONTRIBUTIONS

Name	Student ID	Main Contribution
Li Guanzhen	A0256648N	Development of the hybrid NER model
Wei Pengbo	A0250630U	Development of BERT-based Classifier
Rong Qixian	A0250639B	Data collection and data preprocessing
Lyu Xueyan	A0254414L	Data collection and data preprocessing