Front. Comput. Sci., 2023, 17(3): 173320
https://doi.org/10.1007/s11704-022-1627-2

RESEARCH ARTICLE

APPCorp: a corpus for Android privacy policy document structure analysis

Shuang LIU[1], Fan ZHANG[2], Baiyang ZHAO[1], Renjie GUO[1], Tao CHEN[3], Meishan ZHANG (✉)[2]

1 College of Intelligence and Computing, Tianjin University, Tianjin 300372, China
2 School of New Media and Communication, Tianjin University, Tianjin 300350, China
3 Google, Mountain View, CA 94043, USA

Abstract With the increasing popularity of mobile devices and the wide adoption of mobile Apps, an increasing concern of privacy issues is raised. Privacy policy is identified as a proper medium to indicate the legal terms, such as the general data protection regulation (GDPR), and to bind legal agreement between service providers and users. However, privacy policies are usually long and vague for end users to read and understand. It is thus important to be able to automatically analyze the document structures of privacy policies to assist user understanding. In this work we create a manually labelled corpus containing 231 privacy policies (of more than 566,000 words and 7,748 annotated paragraphs). We benchmark our data corpus with 3 document classification models and achieve more than 82% on F1-score.

Keywords privacy policy, GDPR, document structure analysis, representation learning, graph neural network

1 Introduction

With the rapid development of mobile applications and their wide adoption in different domains, more and more personal data has been provided to application providers. Privacy policy is a document which binds the legal agreement between service providers and users. Therefore, it is fairly important for users to understand the contents of privacy policies before they tick the "I agree" box. For instance, the privacy policy of the ZAO, a Chinese deepfake-like application, explicitly states that the ownership of users' personal data (in particular images uploaded to ZAO) are unconditionally and permanently transferred to ZAO and its affiliates[1)] (excerpt shows in Fig. 1(a)). This term strongly intrudes users' privacy and offends users' right to data rectify, erase and object to processing, as regulated by the General Data Protection Regulation (GDPR) Article 13.2. However, this term hides in the long privacy policy statements and many users of the service were unaware of this term on agreeing the privacy policy.

Reading privacy policies is extremely time consuming. It is reported that each American Internet user needs to spend 244 hours per year to read all the online privacy policies of her visited sites [1]. This is a common, yet hard-to-solve issue due to three reasons. 1) Privacy policies are usually long documents, which are time consuming to read. For instance, on average every privacy policy document has 2,677 words in our dataset. 2) Some privacy policies are poorly structured, or with no structure at all, which makes it harder to read. 3) Privacy policies are usually written with legal terms that are hard for non-experts to understand.

Several studies have been conducted to perform automatic/semi-automatic analysis on privacy policies [2] and human labeled corpora [3,4] are created for the purpose. However, existing work focus on specific and fine-grained aspects of privacy policies, such as vague words [5], detailed information types [4], fine-grained attributes for text segments [3]. None of them attempt to uncover the topics of paragraphs, which could be very useful for outlining or restructuring long documents.

Moreover, regulations have huge impacts on privacy policies [6], especially with the enacting of general data protection regulation (GDPR). For instance, `The New York Times` privacy policy was updated on May 24th, 2018 (one day before GDPR was formally put into action) and a term specifying International Data Transfer, which is explicitly required in GDPR, was added. Figure 1(b) shows the corresponding excerpt of the updated privacy policy. Similarly, 53 out of the 115[2)] privacy policies in the OPP-115 corpus [3] are changed due to GDPR related regulations. A large-scale study [7] found mismatches of the topics extracted from privacy policies (after GDPR is enacted) with the OPP-115 labels. However, none of the existing work considers the legislation impacts on privacy policies.

In this work, we propose a novel task of categorizing the

Received November 4, 2021; accepted February 8, 2022

E-mail: mason.zms@gmail.com

1) The information is extracted from news snapshot, the privacy policy of ZAO has been updated after the report.
2) There are 10 privacy policies which are not accessible.

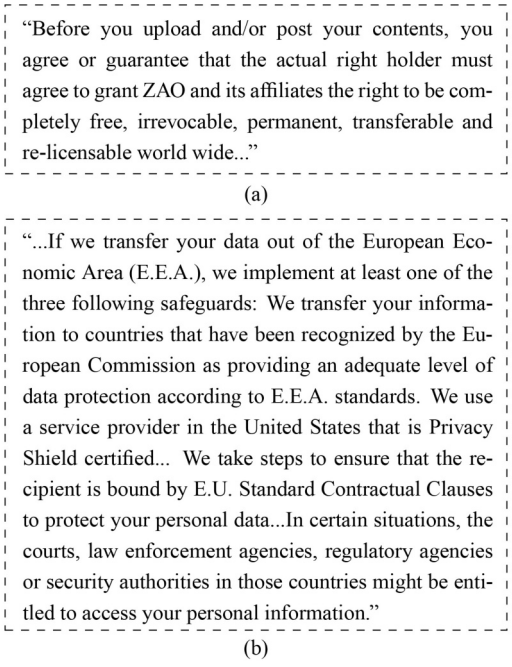

Fig. 1 The privacy policy excerpt examples. (a) The ZAO privacy policy excerpt in English translation; (b) `The New York Times` privacy policy excerpt

topical structures of privacy policies for Android Apps. We devise a classification scheme to characterize the topics for paragraphs with the considerations of related GDPR articles, and then manually label 231 privacy policy documents. We then benchmark the corpus with 3 different model structures, i.e., Support Vector Machine (SVM) [8], Hierarchical Attention Network (HAN) [9] and Hierarchical Graph Attention Network (HGAT), with two different word representations, to learn the paragraph representation from the underlying sentences. The evaluation results show that the HGAT model with BERT as word representation shows the best performance, achieving 83.40%, and 82.50% in terms of Macro-F1 and Micro-F1, respectively.

To the best of our knowledge, this is the first work to involve the GDPR clauses in privacy policy paragraph classification task. Our key contributions include curating the labeled privacy policy corpus, and benchmarking the paragraph classification task with a novel hierarchical BERT model. Our model has downstream applications, e.g., it could be used for outlining privacy policy with paragraph topical labels, and thus helps users to identify the key information from the long document easier. To spur further research, we will make the labeled corpus and our models publicly available.

2 Related work

2.1 Privacy policy corpus creation

Wilson et al. [3] create a privacy policy corpus (OPP-115) of 115 privacy policies, with crowd-sourcing. They develop a policy scheme with ten data practices, which are formulated by domain experts. Each data practice further contains multiple attributes. An arbitrary length of text span is allowed for the annotations. The corpus is labelled in relative fine-grained granularity and only 115 documents are labelled. Our work targets a different task, which outlines privacy policy with paragraph topical labels. Moreover, our label scheme take GDPR into consideration and we labelled 231 privacy policies in our curated corpus, which are obtained after GDPR is enacted. Lebanoff et al. [5] create a vague word (word intrinsic property and is irrelevant with the context) corpus for privacy policies, with the targeting task of vague words prediction. Zimmeck et al. [4] create an app privacy policy corpus (APP-350), targeting the task of app executable and privacy policy compliance checking. They select 350 policies of the most popular apps from the Google Play Store, and hire legal experts to annotate the data. However, their corpus only label a small part of the privacy policy document with fine-granularity. Our labelled corpus targets analysing the whole privacy policy document structure. [10] provides a finer-grained corpus based on OPP-115 [3] with a semi-automatic labelling process, with the focus of opt-out choices in privacy policies. These approaches only cover a small part of privacy policy contents, i.e., APP-350 focuses on particular data types that are collected/shared, [5] targets vague words, [10] and [11] provide labels on opt-out choices and hyperlinks. Our corpus focuses on providing topical labels for paragraphs, and covers a wider range of topics.

There are also approaches conducting semi-automatic labelling in order to reduce human efforts. Liu et al. [12] conduct an empirical study on the problem of aligning or grouping segments of privacy polices. Different approaches, i.e., clustering and HMM are studied and compared with manual labelling from Amazon M-Turk. The results show that an automated approach based on word-level similarities could close about half of the gap between automated approaches and median crowd workers. A recent research [7] conducts unsupervised learning techniques to study the topics in privacy policies and observes that the topics in the privacy policies, after GDPR is enacted, mismatch the topics in the OPP-115 corpus. Tesfay et al. [13] take one step forward to create a corpus including 45 manually labelled privacy policies. The corpus concentrates on the risk levels of the privacy policies defined by experts. Ravichander et al. [14] collect privacy policies for 35 mobile apps from different categories on Google Play and creat a corpus of 1,750 questions and more than 3,500 answers on privacy policies. We focus on privacy policy document structure analysis, and reduce this task into a paragraph classification task. We devise a label scheme based on GDPR articles and also consider the topics discovered by the clustering approach [7] to enrich our label system.

2.2 Automatic privacy policy analysis

Liu et al. [2] utilize logistic regression, support vector machine and CNN model to classify the privacy policy segments and sentences, with the corpus created by [3]. They further split the 'Other' category of the OPP-115 corpus into three categories. i.e., Introductory/Generic, Practice Not Covered, and Privacy Contact Information. The best results show F1 scores of 0.78 and 0.66 for segment and sentence classification, respectively. Kumar et al. [15] train a domain specific word embedding with 300,000 privacy policies. They compare

the classification results (on three models trained with OPP-115) of the domain-specific word embedding with general word embedding using the GloVe ([16]) model. The results show that domain-specific word embedding outperforms general word embedding. Zimmeck et al. [17] propose to combine of machine learning techniques with program static analysis techniques to analyze apps' potential noncompliance with privacy requirements. In particular, they adopt the OPP-115 corpus to train SVM and logistic regression classifiers, which are used to conduct classification on privacy policies. Chang et al. [18] take user profiles into consideration, and automatically list the privacy policy segment description and the corresponding GDPR descriptions which they predict the users are most interested in. The task is reduced to a segment classification task and the TextCNN classifier is adopted and trained on the OPP-115 corpus to classify each segment of the privacy policy. Liu et al. [19] create a corpus of 304 privacy policy documents and conduct a rule-based checking based on 9 rules manually extracted from GDPR Article 13. Our work targets the document structure analysis of privacy policies, and we reduce the task into a paragraph-level classification task, our label scheme considers the impact of GDPR, which the previous work fails to include.

3 Task definition and classification scheme design

3.1 Task definition

We define the privacy policy document structure analysis problem as a paragraph classification task and assign a label, which summarizes the main content of the paragraph, to it. The reason is that privacy policy documents have a relatively fixed set categories of information, which can naturally be summarized by a concise label.

3.2 Classification scheme design

As reported by Degeling et al. [20], the privacy policy contents are largely changed (72.6% privacy policy updates) after GDPR is enacted. Moreover, there is no existing corpus for analyzing the document structures of privacy policy at the paragraph level. Therefore, we propose to create a labelled corpus to help analyze privacy policy structures.

We devise the label scheme based on GDPR articles, i.e., the articles which define the proper actions that should be reflected in the privacy policy[3]. We also take the OPP-115 corpus [3], which is a manually labelled corpus for web application privacy policies, as one of the references in our work. Moreover, we adopt the clustering results by Sarne et al. [7] to help refine our label system. In particular, the International Data Transfer, Policy Contact Information are explicitly labelled in our corpus, which are related to `GDPR Art.13.1(f)` and `GDPR Art.13.1(a)`, respectively. Some other labels, e.g., Cookies and Similar Technologies, are considered in our label scheme due to their importance and the frequent occurrences in Android privacy policies [7].

With the above considerations, we consolidate the topic information [7] summarized through unsupervised learning techniques, the GDPR regulations as well as and expert knowledge (which is mostly consistent with the labels in OPP-115 [3]), and propose a label scheme with 11 labels. We introduce the details of our label scheme in the following:

1. Policy introductory (PI): The general descriptions of the privacy policy document, including definitions on the referential pronouns used in the document.
2. First party collection and use (FPCU): What, when and how the first party (the controller) collects, uses and processes the users' information[4]. [`GDPR Art.13.1`]
3. Cookies and similar technologies (CT): How to collect and use the cookies and other similar technologies (e.g., beacons), and descriptions about those techniques.
4. Third party share and collection (TPSC): How the controller shares and discloses the information with third parties, which include corporate affiliates, service providers or advertising partners.
5. User right and control (URC): The right of user, guaranteed by GDPR and the options which users have in order to control their personal information, such as the settings on users' privacy and safety. For example GPDR requests that the data subject has the right to access, rectify and erase the data. [`GDPR Art.13.2 (b-f)`]
6. Data security (DS): The security facilities/methods that the controller implements to protect users' information. [`GDPR Art.32.1`]
7. Data retention (DR): Descriptions on retention period about users' information. [`GDPR Art.13.2 (a)`]
8. International data transfer (IDT): Descriptions of how is the information stored and transferred internationally. [`GDPR Art.13.1 (f)`]
9. Specific audiences (SA): Specific terms for specific audiences, e.g., children, or data subjects from a specific area/country, which usually has privacy protection laws in power. [`GDPR Art.40.2 (g)`]
10. Policy change (PC): Descriptions on changes of privacy policies and the notification method on changing.
11. Policy contact information (PCI): The contact information of the data controller (i.e., first party). [`GDPR Art.13.1 (a)`]

4 Corpus creation

Our work focuses on the privacy policy of the Android App. Therefore, we collect privacy policies of Apps from the Google Play, one of the most popular Android App stores. We use the Scrapy web framework and Selenium to automates the data crawling process. In our work, we aim at collecting a set of high quality privacy policies with a diverse App categories. Therefore, we follow three strategies to collect the seed links: (1) The privacy policies of Apps which are in the top list of Google Play; and (2) the privacy policies should have a diverse category since different categories may have different

[3] Note that GDPR does not explicitly specify which clauses should be reflected in privacy policies, we consult legal domain experts and select the clauses which are objective and suitable to be reflected in privacy policies in our work.
[4] Note that cookies are usually exempted from the category of user information and regarded as the automatically collected information, which is reflected in Cookies and Similar Technologies.

requirements on accessing user information. The privacy policies we collect cover 22 Google App categories, including Communication, Game, and Business, etc. After obtaining the seed Android App links, the crawler follows the links to locate the corresponding App and then find the privacy policy link.

To ensure the quality of the collected privacy documents, we create the following filtering criteria and the privacy policy documents satisfying all of the criteria are kept: (1) the privacy policy is written in English; and (2) duplicated privacy policies are removed (some Apps from the same company share one privacy policy); and (3) the content of the privacy policy is of reasonable length. We set a minimum size of 2 KB on the privacy policy documents based on observations of average word counts of privacy policies; and (4) the document is describing privacy policy, not some other documents such as Terms Of Services (as some App may put other documents in the link indicating the privacy policy).

We crawl a total of 1,113 privacy policies from Google Play. After the filtering step with the proposed criteria, 231 privacy policies remains. Table 1 shows the statistics of the privacy policies we labelled. There are in total 231 privacy policies labelled, which consists of 7,748 natural paragraphs and 19,708 sentences. There is, on average, 34 paragraphs in each privacy policy document, and each paragraph has around 2.5 sentences and 73 words. We hire a total of 11 annotators, who either major in law or in computer science, and each privacy policy document is labelled by 3 annotators.

4.1 Data annotation

Before annotating the data, we conduct pre-processing on the privacy policy document. We remove noises, such as item symbols, header bars of some web pages, and conduct normalization on links, emails, etc. We also convert all characters to lower case.

We modify the open source labelling tool named YEDDA [21] to include our label scheme and enable our labeling task. In order to properly control the quality of the labelling process as well as the labelled data corpus, we divide our labelling process into two phases. In the first phase, we ask two master students, who work on related research topics, to label 10 privacy policies, and then merge the labels to achieve a consensus, during which a discussion of the initial label scheme as well as the labeling process is conducted based on the issues discovered during the labeling process. We then refine our label scheme and process based on the discussion result. In the second phase, we give a tutorial to all volunteers who are recruited for labelling, which also involve a discussion process to refine the meaning of each label. In total we have 3 volunteers to label each privacy policy. To quantitatively measure how the annotators agree on all the labelled sentences, we compute Fleiss' Kappa [22] inter-annotator agreement. As shown in the last column of Table 2, the Fleiss' Kappa values range from 0.65 to 0.79, which suggests a substantial agreement among the three raters. To further resolve conflicts, we ask the three volunteers to sit together and discuss the conflicted labels that they provid until a consensus is achieved. The whole annotation process takes a total of 25 days.

Table 1 The statistics on the privacy policy corpus

Item	Count
No. Documents	231
No. Sentences	19,708
No. Words	566,475
Annotated paragraph	7,748
Annotators per document	3

4.2 Demographics

In Table 2, `Frequency` is the number of data practices (natural paragraphs in our case) appeared in the corpus. `Coverage` indicates the coverage of the corresponding label, i.e., the percentage of privacy policy documents which contain that label. We can observe that First Party Collection and Use (FPCU) and Third Party Share and Collection (TPSC) count for the majority of the paragraphs. This is also consistent with the findings in the OPP-115 corpus [3]. We can also observe that, some of the labels which are designed based on GDPR requirements, such as User Right and Control (URC) and International Data Transfer (IDT), also appear frequently (61% and 42%, respectively) in the privacy policies labelled. This result indicates that those contents are explicitly documented by many companies (as required by GDPR), and thus the necessity of having those labels.

The `Avg.S` and `Avg.W` indicate the average number of sentences and words for each label. We can observe that most topics have an average of $2-3$ sentences describing the contents. Topics on First Party Collection and Use, Third Party Share and Collection tend to contain more words than other topics. The Policy Contact Information label has the smallest number of words.

Table 2 The per-label statistics in our corpus

Label	`Frequency`	`Coverage`	`Avg.S`	`Avg.W`	Fleiss' Kappa
Policy introductory	638	0.69	2.23	52.92	0.65
First party collection and use	2,433	0.71	2.61	68.45	0.70
Cookies and similar technologies	465	0.48	3.00	64.51	0.72
Third party share and collection	1,316	0.68	2.65	69.90	0.67
User right and control	1,194	0.61	2.39	57.21	0.68
Data security	383	0.62	2.65	59.44	0.79
Data retention	211	0.43	2.26	62.63	0.72
International data transfer	198	0.42	2.39	64.86	0.70
Specific audiences	332	0.57	2.66	67.16	0.78
Policy change	246	0.61	2.80	54.97	0.76
Policy contact information	332	0.65	1.63	30.79	0.70

5 Methodology

Our work targets the task of privacy policy document structure analysis. In particular, we frame the task as a paragraph-level multi-class classification problem, aiming at categorizing each paragraph into the 11 pre-defined topical categories. In this way, we automatically provide a label for each natural paragraph of privacy policies, which could be used for outlining or restructuring long privacy policy documents.

We benchmark the created corpus on the document classification task. In particular, 2 most representative document classification model structures, i.e., SVM [8] and Hierarchical Attention Network (HAN) [9] are adopted. In addition, We propose two orthogonal strategies to incorporate the syntactic knowledge and the contextual word representation to improve the performance of these models.

SVM [8] takes manually crafted discrete features as inputs and has shown to be a strong baseline for a number of NLP classification tasks. Therefore, we adopt it as one baseline of our classification task. We follow the standard settings and use n-gram [23] and tf-idf [24] features in this work.

HAN [9] is specially proposed for the document classification task. The model has a hierarchical attention structure, as shown in Fig. 2(b), which mimics the hierarchical structure of documents, and has shown to be effective in the document classification task. We adopt the HAN's model structure and the bi-directional long short term memory (BiLSTM) [25] as the sentence and document representation model.

The overall architecture of the proposed model is shown in Fig. 2, which mainly consists of four components, i.e., a word encoder, a word-level attention layer, a sentence encoder and a sentence-level attention layer. We describe the details of different components in the following sections.

Input representation Given a document with L sentences and each sentence s_i contains T_i words. w_{it} with $t \in [1, T_i]$ represents the tth word in the ith sentence. We first computes input embedding $\boldsymbol{x}_{it}$ for word w_{it} by concatenating the word embedding $\boldsymbol{w}_{it}$ and its graph embedding $\boldsymbol{g}_{it}$.

For the word embedding, the original HAN model obtains the static word embedding by training an unsupervised word2vec [26] model. In our model, the pre-trained contextual word representation model BERT [27], which has achieved state-of-the-art performance on a wide range of NLP tasks [28], is adopted. In particular, the BERT representation takes one complete sentence as input and output a sequence of hidden vectors at the token level, and each vector has encoded the full-sentence information.

Previous studies [27] usually exploit the outputs of a special token (i.e., `[CLS]`) to represent the input sentence. However, in our model, the embedding of each word $\boldsymbol{w}_{it}$ in the sentence is required in order to incorporate with the graph embedding. Therefore, we adopt the average-pooling over the representations of all tokens corresponding to the word as its representation:

$$\boldsymbol{w}_{it} = \text{AvgPool}(\boldsymbol{t}_{it,1}, \ldots, \boldsymbol{t}_{it,n_{it}}), \tag{1}$$

where n_{it} is the number of the tokens corresponding to the word w_{it}.

In order to capture the syntactic structure of a sentence, we build an N-layer graph encoder by adopting a modified version of GAT [29] to encode its dependency tree. The encoding process computes the graph embedding $\boldsymbol{g}_{it}$ for a word w_{it} based on its word embedding $\boldsymbol{w}_{it}$. To be specific, in the kth layer, the graph encoder computes the output embedding for the word w_{it} as follows:

$$\boldsymbol{g}^k_{\mathcal{N}(w_{it})} = \sum_{w_u \in \mathcal{N}(w_{it}) \cup \{w_{it}\}} \alpha^{k-1}_{it,u} \boldsymbol{W}_r g^{k-1}_u, \tag{2}$$

where $\mathcal{N}(w_{it})$ is a set of words adjacent to word w_{it}. $\boldsymbol{W}_r \in \mathbb{R}^{d_v \times d_z}$ encodes the relation $r \in \mathcal{R}$ between word w_u and w_{it}. The attention coefficient $\alpha^{k-1}_{it,u}$ is computed as:

$$\alpha^{k-1}_{it,u} = \frac{\exp\left(e^{k-1}_{it,u}\right)}{\sum_{w_j \in \mathcal{N}(w_{it}) \cup \{w_{it}\}} \exp\left(e^{k-1}_{it,j}\right)}, \tag{3}$$

where

$$e^{k-1}_{it,u} = \sigma\left(\boldsymbol{a}^\top \left[\boldsymbol{W}_v g^{k-1}_{it} \| \boldsymbol{W}_r g^{k-1}_u\right]\right) \tag{4}$$

is the attention function which measures the importance of adjacent words, considering the edge labels. σ is an activation function, $\boldsymbol{W}_v \in \mathbb{R}^{d_v \times d_z}$ and $\boldsymbol{a} \in \mathbb{R}^{2d_z}$ are model parameters and $\|$ denotes concatenation.

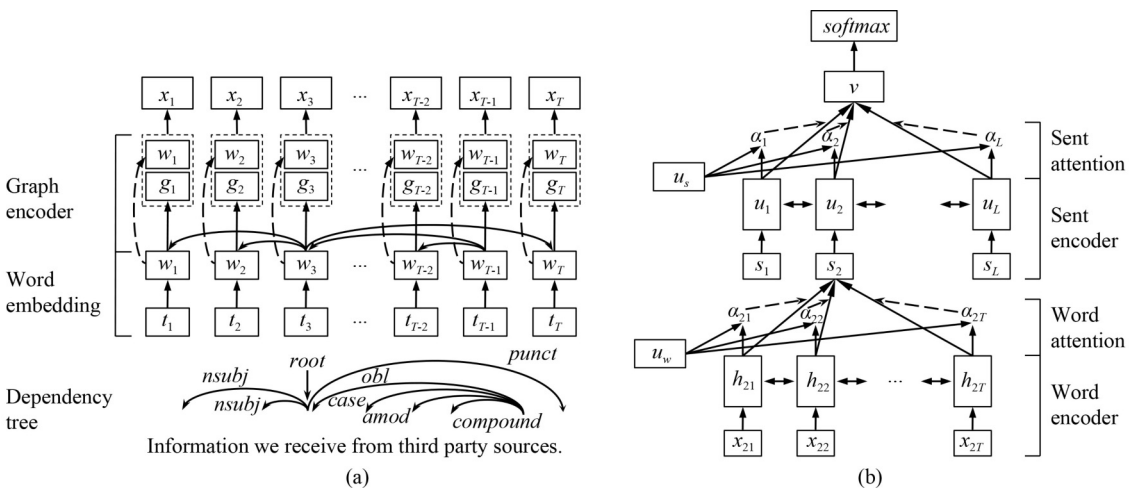

Fig. 2 Input representation and model structure of `HGAT`. The word embedding is adopted from GloVe or BERT. (a) Input representation; (b) the `HAN` model structure

We use multihead attentions to learn distinct relations between words, generating $\tilde{\boldsymbol{g}}^k_{\mathcal{N}(w_{it})}$ by adopting the average-pooling over the M independent heads.

$$\tilde{\boldsymbol{g}}^k_{\mathcal{N}(w_{it})} = \text{AvgPool}(\boldsymbol{g}^{k,1}_{\mathcal{N}(w_{it})}, \ldots, \boldsymbol{g}^{k,M}_{\mathcal{N}(w_{it})}). \tag{5}$$

Then, we use a Gated Recurrent Unit (GRU) [30] to facilitate information propagation between layers and obtain the final representation $\boldsymbol{g}^N_{it}$ as the graph embedding $\boldsymbol{g}_{it}$ for the word w_{it}.

$$\boldsymbol{g}^k_{it} = \text{RNN}\left(\text{g}^{k-1}_{\text{it}}, \tilde{\boldsymbol{g}}^k_{\mathcal{N}(\text{w}_{\text{it}})}\right). \tag{6}$$

Finally, We concatenate the word embedding $\boldsymbol{w}_{it}$ and graph embedding $\boldsymbol{g}_{it}$ to obtain the input embedding $\boldsymbol{x}_{it}$.

$$\boldsymbol{x}_{it} = [\boldsymbol{w}_{it} \| \boldsymbol{g}_{it}]. \tag{7}$$

Encoding and decoding We use a standard bi-directional long short term memory (BiLSTM) to enhance feature composition and feature aggregation. To be specific, a word encoder captures information from both directions for a word w_{it}, and the final representation $\boldsymbol{h}_{it}$ incorporates the contextual information from both directions by concatenating them.

$$\begin{aligned} \overrightarrow{\boldsymbol{h}}_{it} &= \overrightarrow{\text{LSTM}}(\boldsymbol{x}_{it}), t \in [1, T], \\ \overleftarrow{\boldsymbol{h}}_{it} &= \overleftarrow{\text{LSTM}}(\boldsymbol{x}_{it}), t \in [T, 1], \\ \boldsymbol{h}_{it} &= [\overrightarrow{\boldsymbol{h}}_{it} \| \overleftarrow{\boldsymbol{h}}_{it}]. \end{aligned} \tag{8}$$

A word-level attention mechanism is then introduced to measure the importance of each word to the meaning of the sentence and aggregate the representation of those informative words. We compute the similarity of hidden representation $\boldsymbol{u}_{it}$ of $\boldsymbol{h}_{it}$ with a word-level context vector $\boldsymbol{u}_w$ and get a normalized importance weight α_{it} through a softmax function. After that, we obtain a weighted sum of the word representation based on the weights as the sentence vector $\boldsymbol{s}_i$.

$$\begin{aligned} \boldsymbol{u}_{it} &= \tanh(\boldsymbol{W}_w \boldsymbol{h}_{it} + \boldsymbol{b}_w), \\ \alpha_{it} &= \frac{\exp\left(\boldsymbol{u}^\top_{it} \boldsymbol{u}_w\right)}{\sum_t \exp\left(\boldsymbol{u}^\top_{it} u_w\right)}, \\ \boldsymbol{s}_i &= \sum_t \alpha_{it} \boldsymbol{h}_{it}, \end{aligned} \tag{9}$$

where the context vector u_w is model parameter.

Given the sentence vectors $\boldsymbol{s}_i$, we can get a document vector $\boldsymbol{v}$ in a similar way. The document vector $\boldsymbol{v}$ aggregates all the information of sentences in the document and is used as features for document classification. We use the negative log likelihood of the correct labels as training loss. We leave out the details due to space limitations and interested readers are referred to [9] for more details.

6 Experiments

6.1 Data and settings

We divide the corpus into a training set, a validation set, and a test set with the standard partition ratio of 8 : 1 : 1 (based on privacy policy documents) and conduct the standard 10-fold cross-validation to train the models. All the sentences in the corpus are tokenized with StanfordCoreNLP. The `SVM` model is implemented with the scikit-learn [31] package, where we use the linear kernel and set the penalty parameter to be 1.0.

For the static word embedding, we adopt the pre-trained GloVe [16], which is trained on the combination of Gigaword5 and Wikipedia2014 with 6 billion tokens. The vocabulary size is 400,000 and each word is represented by a 100-dimensional vector. For the contextual word embedding, we fine-tune the pre-trained BERT model (BERT-base, uncased) with a learning rate of $2e^{-5}$. In order to compare with GloVe, we first adopt the average-pooling over all representation of tokens regarding a word and feed the output through a one-layer MLP to obtain the word embedding of the same dimension with GloVe. We also use StanfordCoreNLP to obtain the dependency tree of a sentence and implement the graph encoder, with a hidden dimension of 100, 2 modified GAT layers and 4 attention heads, using PyTorch Geometric (PyG) [32]. For the word encoder and sentence encoder in the `HAN` model, we use the 2-layer BiLSTM, with 128 and 256 hidden dimensions, respectively. We set the batch size to be 2 and use Adam [33] as the optimizer. The initial learning rate is set to $5e^{-5}$. We evaluate our models using the F1-score and save the best model on the validation set for testing.

6.2 Main results

The classification results on the test datasets are shown in Table 3, where `P`, `R`, and `F` represent precision, recall and

Table 3 The Precision/Recall/F1 score of classification models

Label	SVM			GloVe						BERT					
				HAN			HGAT			HAN			HGAT		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
`PI`	76.24	69.80	72.88	76.18	73.55	74.84	77.74	78.72	78.23	82.06	77.31	79.61	**82.31**	**79.34**	**80.80**
`FPCU`	75.01	**86.98**	80.55	81.02	82.32	81.66	**83.15**	81.58	82.36	82.55	86.00	**84.24**	82.91	85.02	83.95
`CT`	82.77	73.49	77.85	78.40	78.23	78.32	79.01	79.53	79.27	81.22	80.17	80.69	**83.22**	**81.25**	**82.22**
`TPSC`	78.73	74.83	76.73	79.56	77.80	78.67	77.67	78.26	77.96	**80.57**	80.32	**80.44**	79.48	**80.93**	80.20
`URC`	79.42	76.22	77.79	**81.60**	77.90	79.71	79.87	**81.34**	**80.60**	81.41	77.65	79.48	80.80	78.49	79.62
`DS`	**86.29**	72.51	78.81	77.42	81.68	79.49	82.32	81.68	82.00	82.63	**82.20**	82.41	86.11	81.15	**83.56**
`DR`	86.74	73.71	79.70	74.78	79.34	76.99	78.83	82.16	80.46	81.28	83.57	82.41	**86.96**	**84.51**	**85.71**
`IDT`	**76.06**	83.08	79.41	74.42	82.05	78.05	75.91	85.64	80.48	74.07	82.05	77.86	74.57	**88.72**	**81.03**
`SA`	**92.45**	73.57	81.94	79.83	83.18	81.47	83.58	84.08	83.83	86.08	81.68	83.82	88.12	**84.68**	**86.37**
`PC`	91.60	88.98	90.27	90.72	87.76	89.21	94.64	86.53	90.41	93.28	**90.61**	91.93	**95.67**	90.20	**92.86**
`PCI`	82.37	77.18	79.69	79.41	**81.08**	80.24	**83.02**	80.78	**81.89**	78.92	78.68	78.80	81.08	**81.08**	81.08
`Micro`	78.94	78.94	78.94	79.94	79.94	79.94	80.98	80.98	80.98	81.98	81.98	81.98	**82.50**	**82.50**	**82.50**
`Macro`	82.52	77.30	79.60	79.39	80.44	79.88	81.43	81.85	81.59	82.19	81.84	81.97	**83.75**	**83.22**	**83.40**

F1-score, respectively. We also report both Macro-average and Micro-average scores on all labels for each metric. The SVM, HAN and HGAT models are adopted. GloVe and BERT represent the models using the static word embedding GloVe and contextual word representation BERT, respectively. Results show that the HGAT model with BERT gives the overall best micro-average and macro-average performance on all three metrics. In particular, SVM shows the worst performance among all models except the macro-average precision.

We can observe that the model with syntactic enhancement outperforms the original HAN on all three metrics, which achieves an improvement of 1.71% and 1.43% on the macro-average F1-score for the word embedding GloVe and BERT, respectively. Compared to GloVe, the model with BERT gives a superior improvement of 2.09% and 1.81% on the macro-average F1-score for HAN and HGAT, which reflects the advantages of the contextual word representation containing rich semantic information. Those results demonstrate the effectiveness of the proposed syntactic enhancement and contextual model representation strategies for the HAN model.

We can also see that all models' performances vary in different categories. Some labels, such as international data transfer (IDT), are more complicated than the other labels, and all three models show relatively low prediction results.

6.3 Analysis

6.3.1 Performance analysis on categories

Figure 3 shows the F1-score on different categories. We list the six most representative categories according to the proportion of data items in the test set. We use HAN-GloVe and HGAT-GloVe to denote models using GloVe to encode word representations, and use HAN-BERT and HGAT-BERT to denote models using BERT to encode word representations.

The HGAT-BERT model achieves the best F1-score on 4 out of 6 categories, with the exception of FPCU and TPSC which have no significant difference with the best model. The models using BERT to encode word representations (HAN-BERT and HGAT-BERT) perform better than the models using GloVe to encode word representations (HAN-GloVe and HGAT-GloVe). This is consistent with the reported experiences of using BERT, which has been proven to contain rich linguistic knowledge and semantic knowledge, on other classification tasks.

HGAT-GloVe and HGAT-BERT models are better than all the other models. The deep learning-based models in general outperforms SVM. One exception is the DR label, on which we find that the SVM model can significantly outperform the HAN model and is comparable with that of the HGAT-GloVe model. We check the results of the DR category and find that it has only 211 labelled data items, which could be too small to train a HAN model without any structural or contextual enrichment. Furthermore, the integration of syntactic knowledge makes the model based on GloVe comparable to the model with BERT, e.g., on the label DS.

6.3.2 Performance analysis on paragraph length

Figure 4 shows the F1-score changes with different paragraph lengths, i.e., the number of sentences in a paragraph, and the results are consistent with those in Section 1.

We can observe that HAN-BERT, HGAT-GloVe, HAN-GloVe and SVM show a similar trend with the increase of paragraph length and the best F1-score is obtained with a paragraph length of 3. In particular, the HAN-GloVe and HGAT-GloVe models improve the performance slightly, they perform unstable and drop sharply for paragraph length 4-6, and are worse than SVM in some cases. Enriched with BERT and syntactic knowledge, HGAT-BERT performs the best on all lengths and the performance stops increasing up to a paragraph length of 5, indicating the effectiveness of the two orthogonal strategies proposed. Compared with the models using GloVe for word embedding, the two models using BERT for word embedding have an advantage on paragraphs with multiple sentences, and the advantage is gradually increasing with the increase of paragraph length. One potential reason is that a strong word representation can alleviate the difficulty of encoding long paragraphs for the sentence encoder.

6.3.3 Case study

To further understand the differences between the performance of different models, we explore the effect of the contextual word representation and the syntactic enhancement with a case study.

HAN and HGAT models use an attention-based technique on the sentence level, i.e., it obtains a weighted linear combination of different sentences depending upon their relative importance to the document representation. Figure 5(b) show the visualization of sentence attention weights for an example with the label international data transfer (IDT) which is shown in Fig. 5(a). In these four models, the HAN-BERT and HGAT-BERT models obtain the true results, while the HAN-GloVe and HGAT-GloVe models predict the paragraph incorrectly as third party collection and use (TPCU).

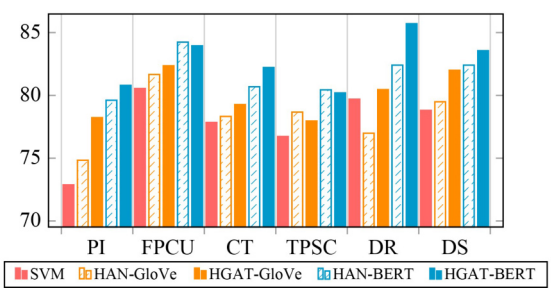

Fig. 3 F1-score against categories

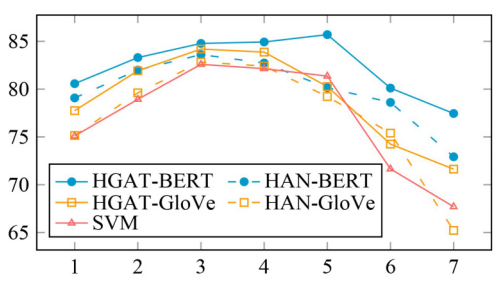

Fig. 4 F1-score against paragraph length

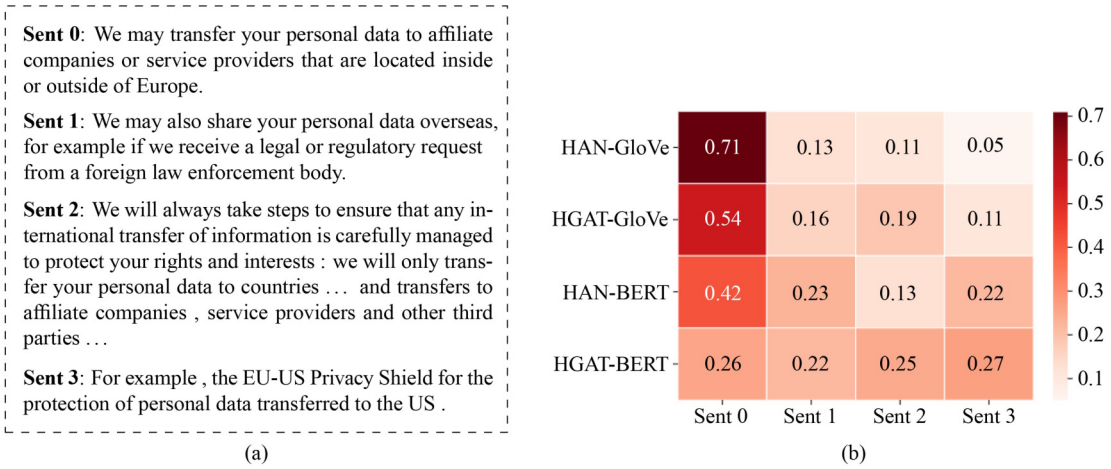

Fig. 5 Visualization of sentence attention for an example from test dataset. The models based on GloVe always give higher attention to the first sentence, while the models based on BERT give higher attention to the more relevant sentences. (a) An example with the label IDT; (b) visualization of sentence attention

We can see that the first two models based on GloVe give very high attention to the first sentence, where the words **"affiliate companies or service providers"** are more related to the label *TPCU*, leading to the wrong classification result.

Compared to BERT, GloVe cannot capture rich semantic information in the last three sentences especially for the OOV word **"EU-US"**, whereas these sentences are also significant to reflect the meaning of the entire paragraph. Although the `HGAT-GloVe` model doesn't predict labels correctly, we can observe that with the syntactic knowledge, the weights become smoother and the model pays more attention to the last three sentences than the `HAN-GloVe` model, indicating the effects of this strategy.

Similarly, we visualize the word attention weights through a heat map in Fig. 6. The paragraph is selected with the label User Right and Control (URC) has only one sentence, and thus the sentence attention is fixed to 1.0 for different models. We also shown the dependency tree of the input sentence.

Only the first `HAN-GloVe` model predicts the paragraph as cookies and similar technologies (CT) incorrectly, because the model is overly dependent on words **"cookie preferences"**, which appear twice in the sentence. Whereas the `HGAT-GloVe` model gives higher attention to the most relevant words according to the syntactic knowledge, especially for the root word **"control"** according to the dependency tree, indicating the effectiveness of the syntactic knowledge.

From the visualization results, We can also observe that the two models with BERT as word representation model give more uniform attention weights compared with GloVe. The finding is consistent with that in the sentence-level attention,

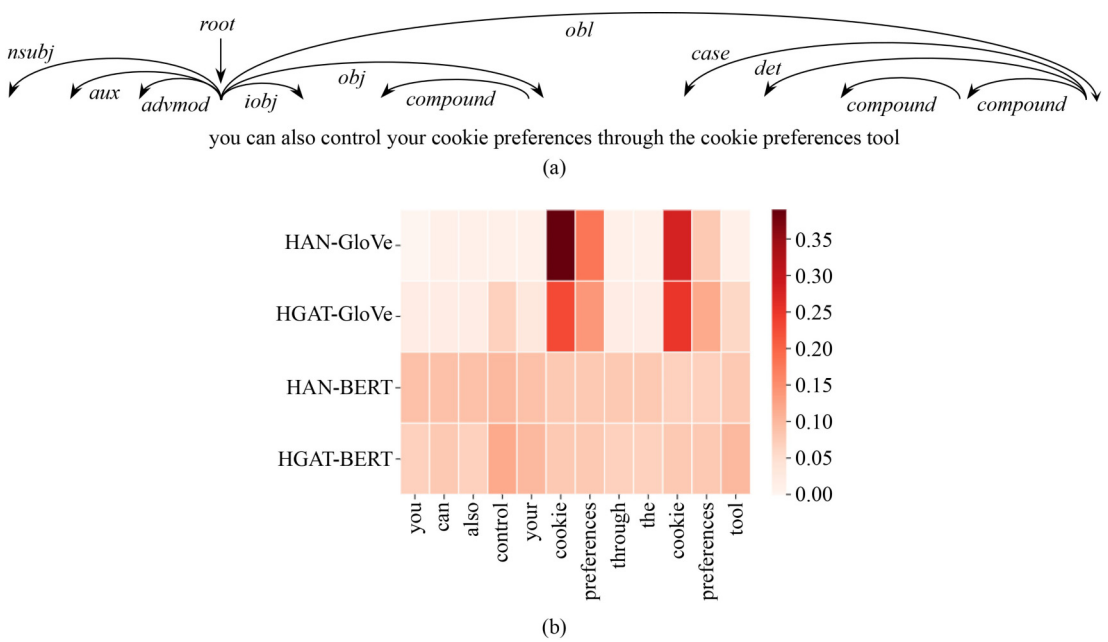

Fig. 6 Visualization of word attention for an example with the label User Right and Control (URC). The models with BERT give uniform attention relatively and the two `HGAT` models pay more attention to the most relevant words according to the syntactic knowledge, especially for the root word **"control"**. (a) The dependency tree of the only input sentence; (b) visualization of word attention

denoting more words are learned well benefit from the power in capturing contextual information in BERT. Furthermore, the best HGAT-BERT model can still pay more attention to the root word **"control"**, demonstrating the effectiveness of the two orthogonal strategies proposed.

7 Conclusion

Automatically analyzing privacy policy is an important problem with the increasing concern on human privacy rights. It is thus very critical to create a high quality corpus to assist this task. In this work, we introduce a privacy policy corpus, which consists of 231 privacy policies. We also benchmark the proposed corpus on the document classification task with 2 widely-adopted models and demonstrate the effectiveness of two orthogonal strategies, i.e., the semantic and structural enhancement. Our model with the enhanced strategies achieves more than 82% on F1-score, which makes the new state-of-the-art. We provide insightful discussions on the evaluation results. Our curated corpus and model have several key applications. Our model can be used to automatically label privacy policies and conduct a large scale of structure analysis. The paragraph label serves as an abstract summary of the corresponding paragraph, which could outline the long privacy policy document, and help users to identify the most important piece of information easier. Since not all privacy policies are well structured, the model could potentially be used for facilitating document restructuring.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 61802275 and U1836214), and the Innovation fund of Tianjin University (2020XRG-0022).

References

1. McDonald A M, Cranor L F. The cost of reading privacy policies. A Journal of Law and Policy for the Information Society, 2008, 4(3): 543–568
2. Liu F, Wilson S, Story P, Zimmeck S, Sadeh N. Towards automatic classification of privacy policy text. Pittsburgh: School of Computer Science, Carnegie Mellon University, 2018
3. Wilson S, Schaub F, Dara A A, Liu F, Cherivirala S, Leon P G, Andersen M S, Zimmeck S, Sathyendra K M, Russell N C, Norton T B, Hovy E, Reidenberg J, Sadeh N. The creation and analysis of a website privacy policy corpus. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016, 1330–1340
4. Zimmeck S, Story P, Smullen D, Ravichander A, Wang Z Q, Reidenberg J, Russell N C, Sadeh N. MAPS: scaling privacy compliance analysis to a million apps. Proceedings on Privacy Enhancing Technologies, 2019, 2019(3): 66–86
5. Lebanoff L, Liu F. Automatic detection of vague words and sentences in privacy policies. In: Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing. 2018, 3508–3517
6. Kaur J, Dara R A, Obimbo C, Song F, Menard K. A comprehensive keyword analysis of online privacy policies. Information Security Journal: A Global Perspective, 2018, 27(5–6): 260–275
7. Sarne D, Schler J, Singer A, Sela A, Bar Siman Tov I. Unsupervised topic extraction from privacy policies. In: Proceedings of 2019 World Wide Web Conference. 2019, 563–568
8. Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995, 20(3): 273–297
9. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: Proceedings of 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016, 1480–1489
10. Sathyendra K M, Wilson S, Schaub F, Zimmeck S, Sadeh N. Identifying the provision of choices in privacy policy text. In: Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing. 2017, 2774–2779
11. Kumar V B, Iyengar R, Nisal N, Feng Y, Habib H, Story P, Cherivirala S, Hagan M, Cranor L, Wilson S, Schaub F, Sadeh N. Finding a choice in a haystack: automatic extraction of opt-out statements from privacy policy text. In: Proceedings of Web Conference 2020. 2020, 1943−1954
12. Liu F, Ramanath R, Sadeh N, Smith N A. A step towards usable privacy policy: automatic alignment of privacy statements. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 2014, 884−894
13. Tesfay W B, Hofmann P, Nakamura T, Kiyomoto S, Serna J. I read but don't agree: privacy policy benchmarking using machine learning and the EU GDPR. In: Proceedings of Web Conference 2018. 2018, 163−166
14. Ravichander A, Black A W, Wilson S, Norton T, Sadeh N. Question answering for privacy policies: combining computational and legal perspectives. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019, 4947−4958
15. Kumar V B, Ravichander A, Story S, Sadeh N. Quantifying the effect of in-domain distributed word representations: a study of privacy policies. In: Proceedings of AAAI Spring Symposium on Privacy-Enhancing Artificial Intelligence and Language Technologies. 2019
16. Pennington J, Socher R, Manning C. GloVe: global vectors for word representation. In: Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. 2014, 1532−1543
17. Zimmeck S, Wang Z, Zou L, Iyengar R, Liu B, Schaub F, Wilson S, Sadeh N, Bellovin S, Reidenberg J. Automated analysis of privacy requirements for mobile apps. In: Proceedings of 2016 AAAI Fall Symposium Series. 2016
18. Chang C, Li H, Zhang Y, Du S, Cao H, Zhu H. Automated and personalized privacy policy extraction under GDPR consideration. In: Proceedings of the 14th International Conference on Wireless Algorithms, Systems, and Applications. 2019, 43−54
19. Liu S, Zhao B, Guo R, Meng G, Zhang F, Zhang M. Have you been properly notified? Automatic compliance analysis of privacy policy text with GDPR article. In: Proceedings of Web Conference 2021. 2021, 2154−2164
20. Degeling M, Utz C, Lentzsch C, Hosseini H, Schaub F, Holz T. We value your privacy... now take some cookies: measuring the GDPR's impact on web privacy. Informatik Spektrum, 2019, 42(5): 345–346
21. Yang J, Zhang Y, Li L, Li X. YEDDA: a lightweight collaborative text span annotation tool. In: Proceedings of ACL 2018, System Demonstrations. 2018, 31−36
22. Fleiss J L. Measuring nominal scale agreement among many raters. Psychological Bulletin, 1971, 76(5): 378–382
23. Wang S, Manning C. Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. 2012, 90−94
24. Ramos J. Using TF-IDF to determine word relevance in document queries. In: Proceedings of the 1st Instructional Conference on Machine Learning. 2003, 29−48
25. Graves A, Jaitly N, Mohamed A R. Hybrid speech recognition with

deep bidirectional LSTM. In: Proceedings of 2013 IEEE Workshop on Automatic Speech Recognition and Understanding. 2013, 273−278

26. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems. 2013, 3111−3119
27. Devlin J, Chang M W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1. 2019, 4171−4186
28. Sun C, Qiu X, Xu Y, Huang X. How to Fine-tune BERT for text classification? In: Proceedings of the 18th China National Conference on Chinese Computational Linguistics. 2019, 194−206
29. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. 2017, arXiv preprint arXiv: 1710.10903
30. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. 2014, 1724−1734
31. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É. Scikit-learn: machine learning in python. The Journal of Machine Learning Research, 2011, 12: 2825–2830
32. Fey M, Lenssen J E. Fast graph representation learning with PyTorch Geometric. 2019, arXiv preprint arXiv: 1903.02428
33. Kingma D P, Ba J. Adam: a method for stochastic optimization. 2017, arXiv preprint arXiv: 1412.6980

Shuang Liu is an associate professor at Tianjin University (TJU), China. She received PhD degree from National University of Singapore, Singapore in 2015. She worked as a research fellow in Singapore University of Technology and Design (SUTD) during 2015−2016, and lecturer in Singapore Institute of Technology (SiT) during 2016−2017. She has been a faculty member of TJU since 2018. Her research interests focus on software quality assurance and privacy protection in general.

Fan Zhang received his BS degree from Northeastern University (NEU), China in 2020 and is currently a master student in Tianjin University (TJU), China. His research interest is natural language processing (NLP). He is currently working on pre-trained language models.

Baiyang Zhao received his BS degree from Dalian Maritime University (DMU), China in 2019 and received his MS degree from college of Intelligence and Computing, Tianjin University (TJU), China in 2022. His research interest is natural language processing (NLP), privacy protection and pre-trained language models.

Renjie Guo received his BS degree and MS degree from College of Intelligence and Computing, Tianjin University, China. His research direction is automatic analysis of privacy policies based on natural language processing technology, especially the semantics and classification of privacy policy paragraphs.

Tao Chen received the PhD degree in Computer Science from National University of Singapore, Singapore in 2016. She is currently a research engineer in Google Research, Mountain View, USA. Prior to that, she was a postdoc in Johns Hopkins University, USA. Her research interests lie in natural language processing, information retrieval, health informatics, social computing and multimedia.

Meishan Zhang received his PHD degree from Harbin Institute of Technology (HIT), China in 2014. He is now an associate professor in School of New Media and Communication, Tianjin University, China. His major research interests include natural language processing and machine learning.