Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

AI-enabled legacy data integration with privacy protection: a case study on regional cloud arbitration court

Jie Song[1], Haifei Fu[1], Tianzhe Jiao[1] and Dongqi Wang[1*]

Abstract

This paper presents an interesting case study on Legacy Data Integration (LDI for short) for a Regional Cloud Arbitration Court. Due to the inconsistent structure and presentation, legacy arbitration cases can hardly integrate into the Cloud Court unless processed manually. In this study, we propose an AI-enabled LDI method to replace the costly manual approach and ensure privacy protection during the process. We trained AI models to replace tasks such as reading and understanding legacy cases, removing privacy information, composing new case records, and inputting them through the system interfaces. Our approach employs Optical Character Recognition (OCR), text classification, and Named Entity Recognition (NER) to transform legacy data into a system format. We applied our method to a Cloud Arbitration Court in Liaoning Province, China, and achieved a comparable privacy filtering effect while retaining the maximum amount of information. Our method demonstrated similar effectiveness as the manual LDI, but with greater efficiency, saving 90% of the workforce and achieving a 60%-70% information extraction rate compared to manual work. With the increasing development of informationalization and intelligentization in judgment and arbitration, many courts are adopting ABC technologies, namely Artificial intelligence, Big data, and Cloud computing, to build the court system. Our method provides a practical reference for integrating legal data into the system.

Keywords Legacy data integration, Privacy filtering, AI-enabled, Cloud court, Natural language processing

Introduction

With the continuous development of cloud computing and the emergence of numerous cloud service providers, various fields, such as business, education, and governance, have started migrating their traditional systems to the cloud to provide more efficient and convenient services at lower costs [1]. In China, an increasing number of courts and arbitration tribunals have built their "digital justice systems" through digital techniques and cloud services, namely *Cloud Courts*, to improve the efficiency of legal proceedings and promote the fairness and openness of the judicial process [2]. A Cloud Court typically includes court websites, case trial processing systems, information and status of court cases, case management tools, etc. In the legal field, historical data accumulated over time holds significant reference value for future case hearings and arbitration processes. Therefore, integrating legacy data is unavoidable when constructing a Cloud Court. Whatsmore, privacy protection in such Legacy Data Integration (LDI for short) is also a prioritized issue that must be addressed well [3].

As shown in Fig. 1, the obstacle to constructing a new cloud system is the effective integration of big and valuable legacy data into the Cloud Court. Despite the availability of various cloud storage services such as relational database services, object storage services (OSS), data warehousing services, and multiple data loading

*Correspondence:
Dongqi Wang
wangdq@swc.neu.edu.cn
[1] Software College, Northeastern University, Shenyang, China

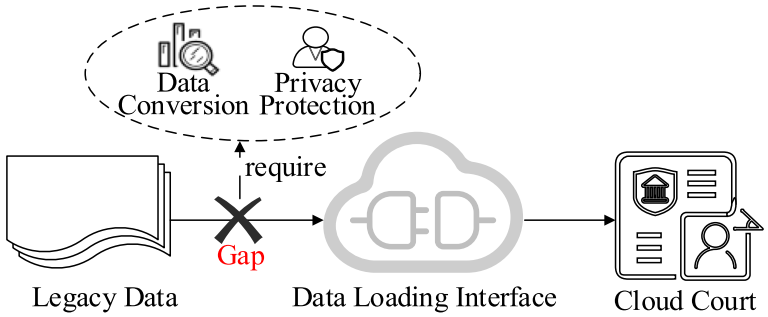


Fig. 1 Legacy data integration in cloud court

methods, the process of LDI remains a significant challenge that cannot be overlooked during the migration to the cloud. The coarse-grained data integration interfaces provided by cloud service providers may not adequately address the finer-grained data integration required by traditional systems. Additionally, the use of public cloud infrastructure presents potential risks to data privacy [4], which makes it necessary to employ additional efforts to ensure privacy protection.

The convergence of cloud computing and artificial intelligence has led to an increasing number of studies aimed at solving problems in cloud computing with AI technology [5–7]. Against this backdrop, Our research aims to propose an AI-enabled LDI method that reduces labor costs during the integration and provides a feasible solution to extract information from unstructured text data with privacy protection. To achieve this, we combine structured and Natural Language Processing (NLP) based unstructured extraction methods to transform historical documents containing structured and unstructured text data into structured form. After obtaining structured data, we apply anonymization and NLP techniques to achieve privacy filtering before integrating the data into the Cloud Court database. The practice of integrating legacy data at a Regional Cloud Arbitration Court in China has demonstrated the effectiveness of our method in reducing labor costs. Besides, the case's extraction and privacy protection results are comparable to the manual ones. We call the illustrated case the AI-enabled LDI with privacy protection for the regional cloud arbitration court. Through the case demonstration, this paper makes the following contributions.

1) We propose an NLP-based AI-enabled LDI method to address the problem of integrating unstructured text data in the arbitration area which significantly reduces manual labor costs.
2) We have incorporated privacy protection considerations into the LDI process and achieved a certain degree of privacy protection using anonymization and NLP techniques, without relying on manual intervention.
3) We have demonstrated the feasibility of our proposed method through its successful application in the practice of LDI in the region.

The paper is organized as follows: Related works section introduces related work. Preliminary section introduces the architecture, legacy data, and database of regional cloud arbitration court LDI tasks; and then overviews previous manual LDI solutions. AI-enabled legacy data integration and AI-enabled privacy protection sections describe the legacy data integration and privacy protection methods. Functional evaluation and Human-AI comparison sections evaluate the effectiveness of our proposed method from the perspectives of functional evaluation and human-AI comparison. The last section summarizes our work and outlines future research.

Related works

This section focuses on two aspects: data integration and privacy protection. Based on a comprehensive analysis of related research, we summarize the similarities and differences between our study and prior work.

Data integration

The primary objective of a conventional data integration process is to transform data originating from multiple sources into a target representation [8], typically involving three steps: schema mapping, duplicate detection, and data fusion. This field focuses primarily on integrating database tables, including case studies on various domains. Leng et al. [9] investigate BIM-GIS (Building Information Modeling, Geographic Information System) integration issues in site planning and propose a comprehensive integration scheme that involves three stages: extraction, integration, and optimization, specifically targeting BIM/GIS data sets stored in *JSON* and *obj*

object formats. Reda et al. [10] propose a layered integration scheme for health and fitness data, transforming metadata into uniformly formatted data based on the IoT Fitness Ontology. In the cloud environment, data integration originates not only from traditional systems but also from various types of other cloud-based application data. S. K. Habib et al. [11] implemented IoT data integration middleware by incorporating Representational State Transfer (REST) paradigms. N. Prasath et al. [12] proposed a method for migrating data from multiple sources to cloud storage using the Extract Transform Load (ETL) tool.

Through the analysis of the above studies, current data integration efforts mainly focus on integrating structured data, such as database tables, and semi-structured data, such as *JSON*; and primarily address the problem of transforming data across different structures into a unified representation. Moreover, the domains and data objects targeted by various data integration studies make it challenging to directly apply their findings to the data integration scenario examined in this paper.

There has been considerable research in the field of information extraction on how to obtain target information from unstructured data. Rodriguez et al. [13] proposed the FADOHS framework based on sentiment and emotion analysis techniques, achieving promising results in integrating unstructured data on hate speech on Facebook. Liu et al. [14] proposed a pattern-based approach to extract disease and drug combination pairs from MEDLINE abstracts. Nguyen et al. [15] utilized the NLP model Transformer to extract information from domain-specific business documents with limited training data. M. Kerroumi et al. [16] proposed a multimodal approach, VisualWordGrid, to extract information from documents with rich visual characteristics, such as tables. Through the literature review, it becomes apparent that extracting information from unstructured data often necessitates AI-based methods. In contrast, the traditional pattern-matching methods adopted in conventional data integration processes may not be suitable for integrating unstructured data.

In summary, our study employs AI-based data extraction to address the data integration issue in our application scenario. It proposes a data integration scheme with both pattern matching- and AI-based information extraction.

Privacy protection

Ensuring the privacy protection of legacy data during integration is a key focus of many case studies, and encryption is a common method in public cloud databases to ensure cloud data security. S. Liu et al. [17] implemented a middleware that provides users with transparent querying on different encrypted databases in the public cloud. A. A. Alqarni [18] proposed Paillier Homomorphic Encryption to encrypt cloud data and decrypt it on the user end to ensure data security. In our application scenario, only a small portion of the data is sensitive, and encryption would consume excessive cloud computing resources.

Privacy-Preserving Data Publishing (PPDP) has four common strategies for handling privacy data: generalization, anatomy, perturbation, and suppression. In privacy-preserving data publishing, *k*-anonymity, *l*-diversity, and *t*-closeness are classical privacy protection algorithms. Ren et al. [19] published graph data securely by applying *k*-anonymity and *t*-closeness techniques. Khan et al. [20] proposed a "*single identity clustering privacy filtering method*" based on *t*-closeness and validated its efficacy on healthcare privacy data.

However, the above privacy protection algorithms are primarily designed for relational databases and may not fully address the privacy concerns of unstructured text data. Therefore, we consider privacy-preserving natural language processing techniques. Iwendi et al. [21] proposed a comprehensive method that utilizes regular expressions and the Stanford Named Entity Recognition Tagger to sanitize sensitive information in healthcare documents. Moqurrab et al. [22] improved the effectiveness of medical entity recognition using deep neural networks for unstructured big biomedical data. Zheng Li et al. [23] presented an anomaly detection framework that utilizes an attention mechanism in deep learning to reduce the considerable computing power and resource requirements of the detection process. To this end, the *t*-closeness and NLP-based filtering methods are our application's preferred privacy protection for legacy legal data. Our method combines the strengths of PPDP techniques with NLP-based methods to address the privacy concerns of unstructured text data.

Preliminary

In this chapter, we present a brief discussion on the functionalities, runtime data, and legacy data of the Regional Cloud Arbitration Court. Additionally, we present the manual LDI and privacy filtering methods, which serve as a foundation for understanding our AI-enabled LDI method.

System architecture

The system architecture of the Reginal Cloud Arbitration Court system is shown in Fig. 2. The annotations in Fig. 2 are self-described, so the detailed explanation is abbreviated here. However, the points related to our studies are highlighted here:

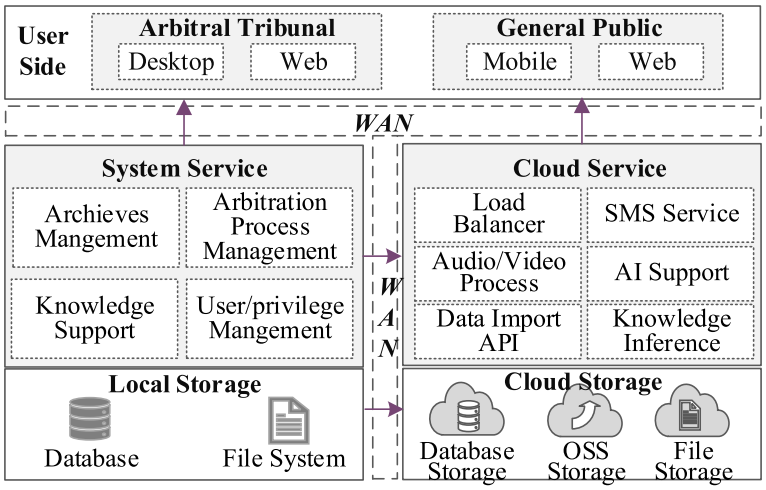


Fig. 2 Software architecture

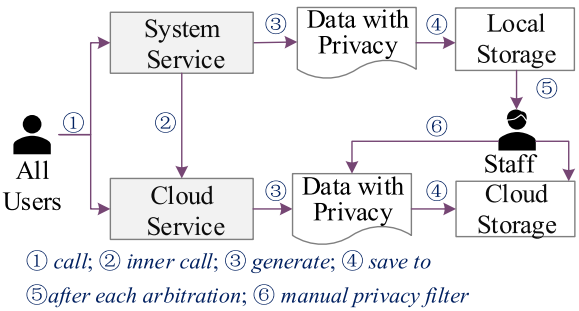


Fig. 3 Data flows in the storage

- The RESTful interfaces are for the client-side application to access the cloud services. They are designed for automatic LDI, while manual one is via GUIs(Graphical User Interfaces).
- The Knowledge Support components provide users with a knowledge query on the knowledge graph: correlation among entities such as cases, laws, crimes, and penalties extracted from arbitration cases.
- The local database is deployed on the internal server of the arbitral court and is responsible for storing the original arbitration data with private information. In contrast, the cloud database stores the masked arbitration data and the knowledge graph.

Data flows

As shown in Fig. 3, arbitration data is stored in the local database during the arbitration process; After arbitration, masked arbitration data are obtained by *m* privacy removal of the current arbitration case in the local database. Then the masked arbitration data is uploaded to the cloud database.

The LDI process is directed towards the legal document data accumulated by the arbitration court, which is extracted and transformed into a structured database format, and subsequently imported into the cloud database.

Legacy data

Legacy data refers to documents recording historical arbitration cases. Legacy data has three forms: electric documents and tables, printed documents with typed words, and paper documents with handwritten words. These documents are managed case by case, and the paper documents are scanned and stored in the local file system. Table 1 shows the general structure of documents arbitration cases required.

Cloud database

Figure 4 lists the columns, tables, and references of the structured cloud database. There are 15 entities (tables) and 17 relationships between entities. These tables are the "destination" of LDI, where most columns are extracted from the legacy data files. The details of the tables are abbreviated.

Manual LDI

The manual LDI process generally has three stages. *First*, the analysts define the extraction rules, namely the relationship between legacy data files and cloud databases, and train the staff to grasp these rules. *Second*, as human information processors, the staff extract key information from the legacy data files according to the rules and load it into the cloud database through a batch-loading tool. This phase is the most time-consuming and

Table 1 Document structure for an arbitration case

No	File Name	Page Number
1	Case Acceptance Approval Form	1–1
2	Arbitration Application Form	2–2
3	Copy of Applicant's ID Card	3–3
4	Identity Certificate of Attorney	4–4
5	Copy of Respondent's ID Card	5–5
6	Copy of Applicant's ID Card	6–10
7	Evidence Submitted by Applicant	11–15
8	Evidence	16–17
9	Confirmation Info	18–18
10	Court Record	19–21
11	Arbitration Award	22–23
12	Copy of Arbitration Award	23–24
13	Service Return Receipt	25–26
14	Appendix	27–29

labor-intensive phase of manual data integration. *Last*, the integration results of individuals are sampled and checked respectively.

In past practices of manual data integration, several encountered problems often led to poor results in manual data integration.

- **Laborious manual integration.** Many legacy documents have long text content, and the integration work requires the staff to understand the documents thoroughly.
- **Work efficiency problem.** According to the practical experience of Manual LDI, as the integration work progresses, staff may be tempted to disregard the integration rules to reduce their workload, resulting in a poor integration effect.
- **Differences in the understanding of LDI rules.** It is difficult for staff to fully understand the integration rules, which leads to the failure to achieve the expected data integration effect.

Manual privacy filtering

Manually filtering data privacy before uploading to the cloud database is essentially the same as manual LDI. However, such a data process presents two issues:

- **Roughness.** Manual privacy filtering is rough and does not consider the trade-off between data quality and privacy filtering. The staffs merely process the privacy data items that satisfy the simple filtering rules, without comprehensively considering the filtering strategy for data items from the perspective of the overall data distribution and the impact of privacy filtering on data quality.
- **Privacy leakage.** Despite the imposition of strict access controls, privacy breaches may still occur during the privacy filtering process, as it is inevitable that staff members will need to access sensitive data.

AI-enabled legacy data integration

As shown in Fig. 5, the AI-enabled LDI includes three stages: *Paper Document Conversion*, *Information Extraction*, and *Data Integration*, which implements the conversion from paper documents to database structure data. This section analyzes each stage's processing techniques and methods, combined with data examples.

Paper document conversion

The paper document conversion phase transforms the original paper document into program-readable text data. As shown in Fig. 6 (virtual data in Chinese), the original paper documents can be divided into three categories according to their writing form:

- **Printed**. These paper documents are printed copies of electronic versions. Documents of the same type have a unified and standardized format, and the font is clear and recognizable. This type of document can be easily digitized and extracted.
- **Handwritten**. This kind of document comes from earlier arbitration cases, and the relevant people handwrite the contents. Diversified writers may significantly affect the fonts, format, writing style, and expression of the same document type.
- **Mixed**. These paper documents mix the printed and handwritten contents. They are more unambiguous and more distinguishable than the handwritten ones. They are commonly printed forms and statements filled or extended by relevant people's handwriting.

In this stage, the handwritten type is manually distinguished from the other two types, and a scanner is adopted to obtain the text image. After the classification and scanning are completed, the Optical Character Recognition (OCR) script based on Open Source OCR Tool Tesseract is called to convert text images into text data in batches, and two types of output are obtained according to the input classification: the well-formatted data and the poorly-formatted data. The former has a standardized and unified format from the OCR

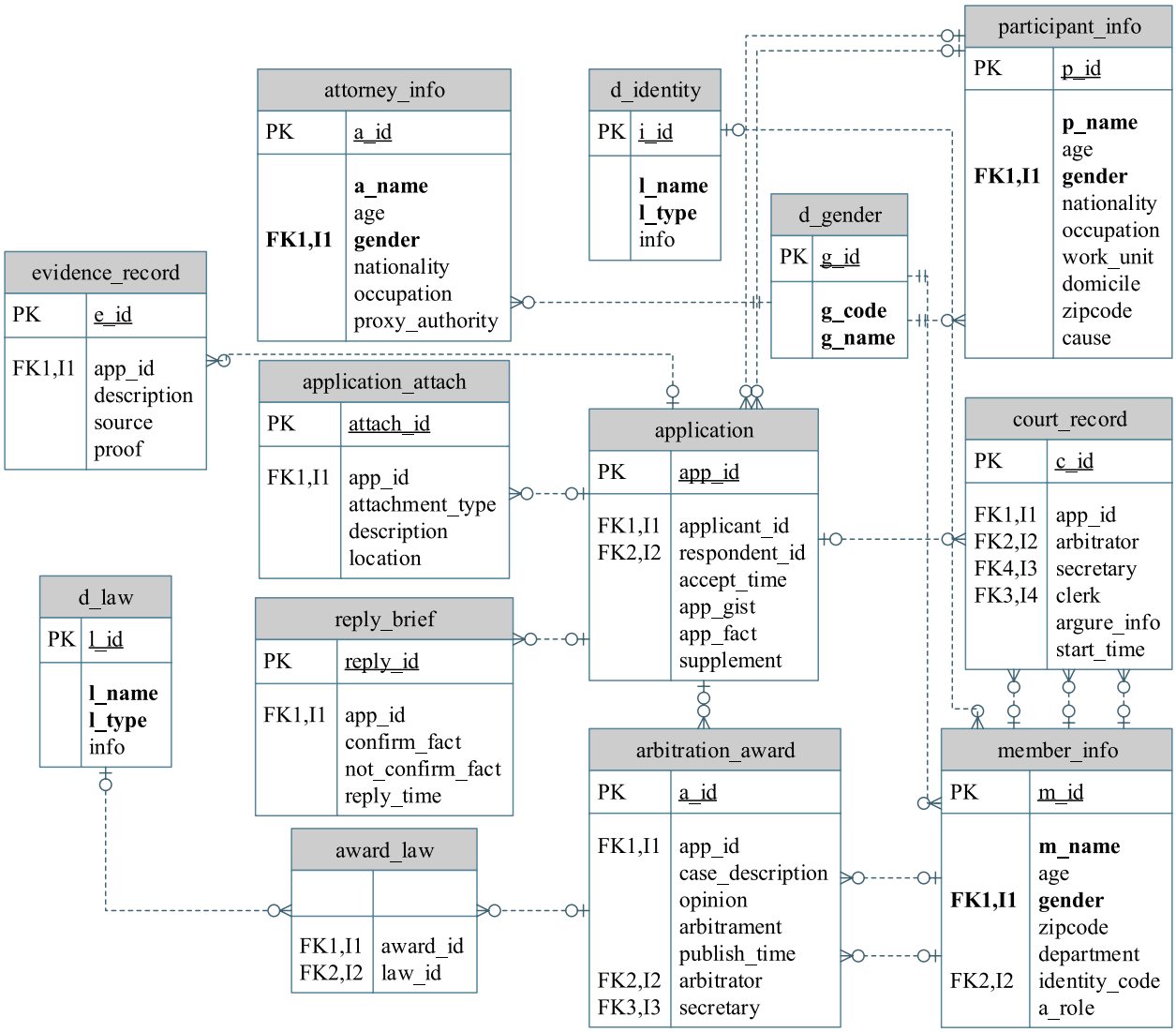


Fig. 4 Database schema of the regional cloud arbitration system

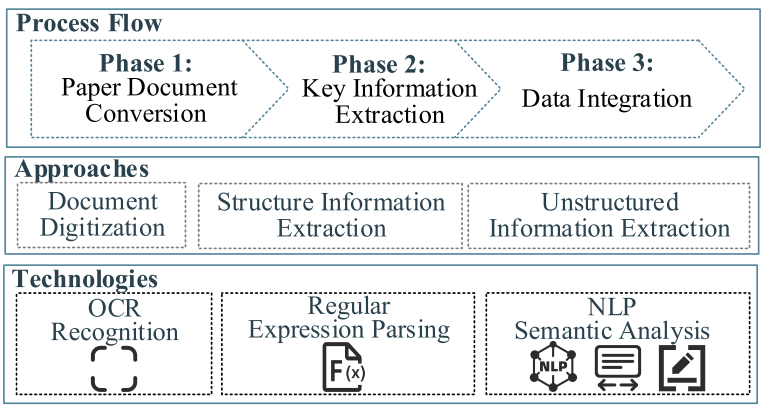
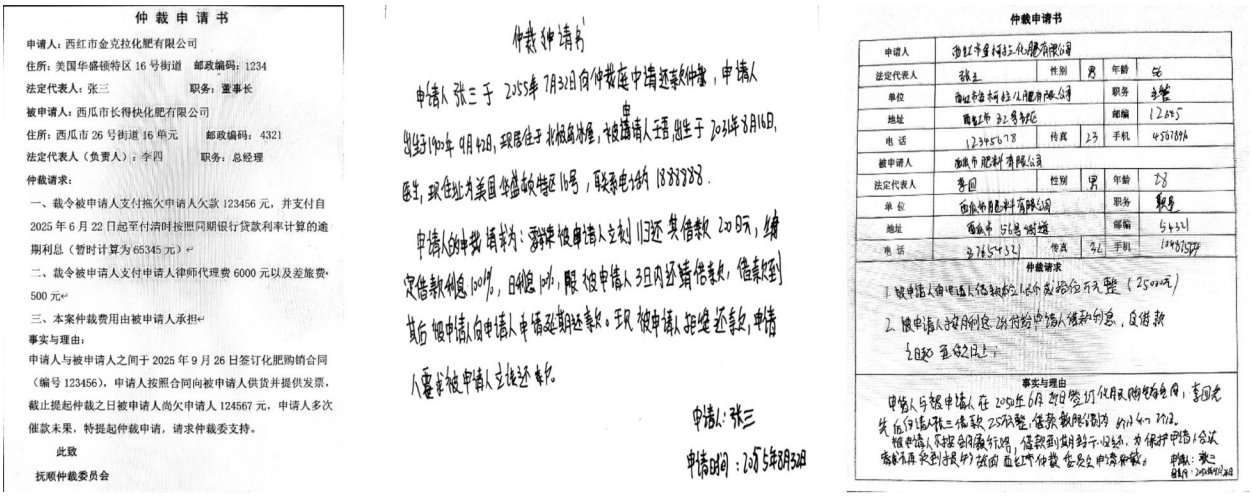


Fig. 5 AI-enabled LDI overview



(a) printed (b) handwritten (c) mixed

Fig. 6 Example of different paper document formats (virtual data in Chinese). **a** Printed paper documents. **b** Handwriting paper documents that vary in format. **c** Mixed format paper document that contains both print and handwriting content

results of printed documents. The latter, which pertains to handwritten and mixed documents, exhibits a freestyle and unified expression. It consists of multiple paragraphs, and the sentences in each paragraph are directly related. It cannot be extracted using similar methods as the former.

Key information extraction

As shown in Fig. 7, the key information extraction stage utilizes distinctive techniques to extract information from the two data types acquired in the previous stage, subsequently transforming them into structured data in JSON format. Before delving into a detailed description of our extraction method, it is essential to introduce a fundamental concept:

Definition 1. Segment Group (SG). A Segment Group (SG) refers to a set of semantically related sentences that appear consecutively within the source text data. An SG exhibits a similar structure and possesses a robust semantic correlation, with the information extracted from an SG corresponding to a data table or a subset of relevant fields within a table. Semantic analysis and information extraction are conducted on an SG-by-SG basis during the information extraction phase.

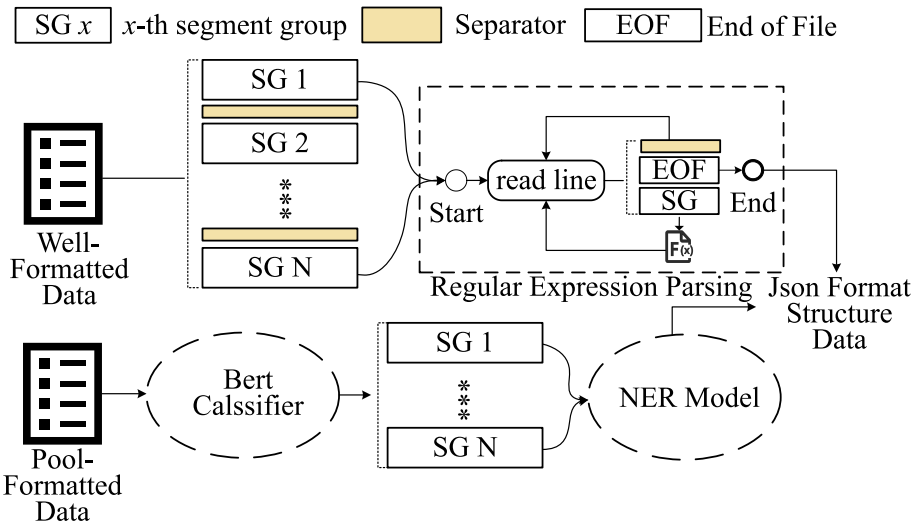


Fig. 7 Detailed extraction process

Well-formatted data extraction

Figure 8 is the well-formatted data example extracted from Fig. 6(a) (translated in English). The unified and standardized features of well-formatted data are reflected in three aspects, and the latter two aspects make the regular expressions-based extraction method feasible.

- Fewer recognition errors and wrong words reduce the information extraction difficulty.
- Clear content structure and boundaries make SGs recognizable. For example, SGs of the arbitration application, such as personal information and application request, are separated by a title line.
- Key information, such as personal information at the beginning of the arbitration application, is as structured as the key-value pair.

Considering the above reasons, we adopt a rule-based information extraction method, such as regular expressions, to extract this type of data. Algorithm 1 shows how the extraction script reads the input file line by line and extracts key-value pairs using a predefined regular expression. The process is repeated until all rows are parsed. For instance, the applicant information block in the arbitration application contains the applicant's name, gender, and nationality after the word "*applicant*" and is separated by commas.

Input: For input with n lines and m section, Inputs include all lines $L = l[1], l[2], \ldots, l[n]$ and $S = s[1], s[2], \ldots, s[m]$ section separator and regular expressions $REs = re[1], re[2], \ldots, re[m]$
Output: The JSON format information extraction result

```
ExtractRes()
1   extractRes ←empty JSON object //initialize output
2   secIndex ←1//current section index
3   secContent ← empty string //current section content
5   for i = 1 to n do
6       if i==n // current line is the last line
7           extractRes.add(re[i].parsing(secContent))
8       else if  l[i] = s[secIndex] // current line has separator
9           extractRes.add(re[i].parsing(secContent))
10          secIndex ←secIndex + 1// increment section index
11          secContent ← empty string //clear section content
12      else // current line has no separator
13          secContent ←secContent + l[i]
14      end else
15  end for
16  return extractRes
```

Algorithm 1. General extraction script based on regular expression

As shown in Code 1, the extracted results of the arbitration application include personal information of the applicant and respondent, arbitration request, and arbitration facts, which are stored in JSON format.

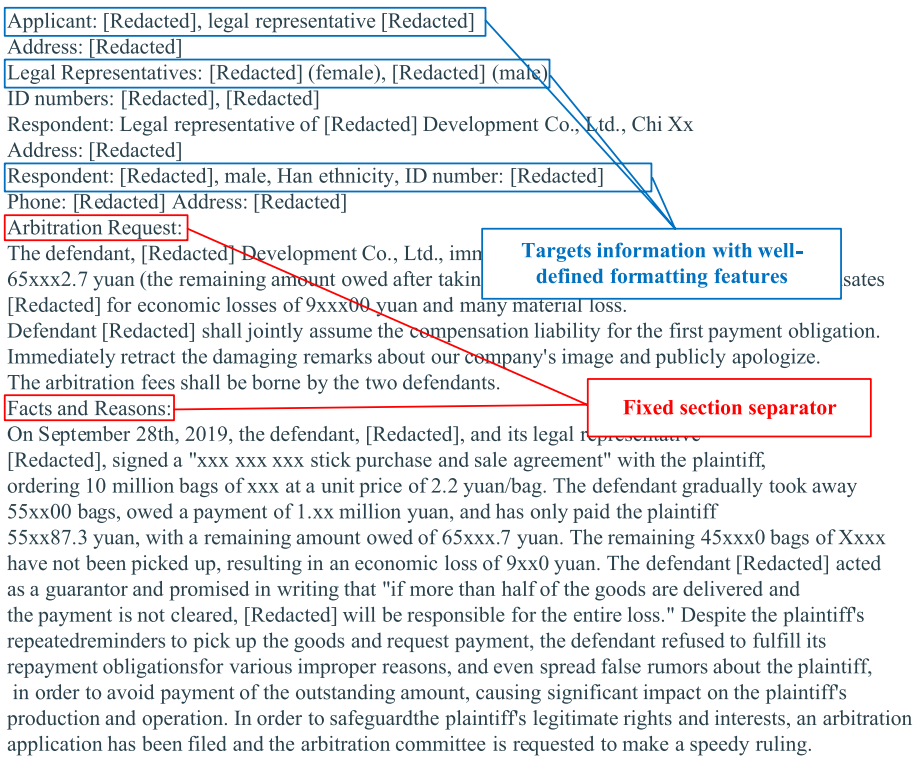


Fig. 8 Well-formatted data example

```
{
  "Arbitration Applicant": {
      "Name": "Zhang San",
      "Gender": "Male",
      "Ethnicity": "Han",
      "Education Level": "High School",
      "Occupation": "Unemployed",
      "Date of Birth": "August 18th, 1976"
  },
  "Respondent": {
      "Name": "Li Si",
      "Gender": "Female",
      "Ethnicity": "Han",
      "Education Level": "High School",
      "Occupation": "Unemployed",
      "Date of Birth": "August 18th, 1986"
  },
  "Arbitration Request": "Requesting a legal ruling that the respondent
                         pay 365,310 yuan .....",
  "Facts and Reasons": "On August 28th, 2008, the applicant and the
                       respondent entered .....",
  "Application Time": "September 4th, 2011"
}
```

Code 1. Example JSON for extracted results

Poorly-formatted data extraction

In contrast to well-formatted data, poorly-formatted data exhibits a random structure and expression, lacks coherent organization and discernible separation between its various content blocks, and contains typos and recognition errors in text content. In this case, the regular expression parsing is unworkable. We employ a comprehensive AI method to extract information. It has two phases: content partition and Named Entity Recognition (NER), which involve the segmentation and identification of key information within sentences.

Content partition

We adopt the pre-training model BERT(Bidirectional Encoder Representation from Transformers) [24] in the content partition phase to segment the document's content. The document is divided into multiple semantic groups based on the desired information extraction requirements, and further processing proceeds accordingly. The following outlines the specific data flow of this phase.

In our approach, we treat the partitioning of text content as a classification problem. We first define the classification labels for each SG of different text data types. For instance, we identify eight SG labels for arbitration document data denoted by l_1 = '*arbitrator information*', l_2 = '*arbitration case description*', ..., l_8 = '*arbitration result*'.

Following the fine-tuning of the BERT model, tokenization is performed on each sentence S, with the [CLS] token added to the beginning of the sequence. The resulting sequence is then fed into the BERT model, with the final hidden vector C_H of the [CLS] token serving as the sentence representation. This representation is then projected onto the classification label space using a fully connected layer. By applying a softmax function, we obtain the subgroup (SG) label l_S that has the highest probability and assign it as the classification label for the input sentence. The calculation method is presented in formulas (1) and (2) below:

$$y^{'} = softmax\left(W^T C_H + b_L\right) \tag{1}$$

$$l_S = max\left(y_i^{'}\right), i = 1, 2, 3...L \tag{2}$$

where W_{L*H}^T, b_L represent the weight and bias of the fully connected layer, H is the hidden layer dimension, and L denotes the number of SG labels of current document type. By applying this step, we separate the document data consisting of N sentences, i.e., $\{S_1, S_2, \ldots, S_N\}$, into a set of *SG*s denoted as $\{SG_1, SG_2, \ldots, SG_L\}$. The variable L denotes the total number of SGs that are associated with the document type.

Named entity recognition (NER)

In the phase of NER, we perform entity recognition on the divided test data based on the NER model, so-called BERT-BiLSTM-IDCNN-CRF [25], and return the target entity as the key information of the current text extraction.

We selected both well-formatted data and poorly-formatted data for entity annotation training. Well-formatted data was annotated using regular expressions. Poorly-formatted data were manually labeled. All labeling work is conducted using the BIO method, which employs B (Begin), I (Interior), and E (End) labels to distinguish between different words within a single entity. As shown in Table 2, we have defined around 20 named entities according to the data requirements. For each entity type, we constructed around 500–1000 training data instances from the source documents.

In the preceding step, we separated the document data into a set of SG denoted by $\{SG_1, SG_2, \ldots, SG_L\}$. In this step, the data is processed at the SG level. Specifically, each sentence within a subgroup is traversed and tokenized. The resulting tokens $X = \{x_1, x_2, \ldots, x_n\}$ are then input into the BERT and BiLSTM-IDCNN layers to obtain sequence features, denoted by $H = \{h_1, h_2, \ldots, h_n\}$. Subsequently, the score of each NER tag for each word is computed through linear mapping:

$$P_i = w_{k*n} h_i + b_k \tag{3}$$

where w_{k*n} and b_k are linear mapping parameters, k is the number of NER labels, and P_i is the score of the i-th token for the corresponding NER tag. Afterwards, the scores are input into the CRF layer to calculate the transition score, which can be expressed as follows:

Table 2 Named entity definitions

Named Entity Group	Named Entities	Train	Validate	Test
Applicant/ Respondent	gender(A_GEN), age(A_AGE), ID(A_ID), occupation(A_OCC), nationality(A_NAT)	1200	200	232
Court Record and Arbitration Award	attorney(C_ATO), arbitrator(C_ARB), secretary (C_SEC), clerk(C_CLE), law(C_LAW), case code(C_COD), case reason(C_CAS)	1300	150	123
Evidence Detail	evidence(E_EVI), verified(E_VER), discussion(E_DIS)	200	30	30
General Entity	date(G_DATE), location(G_LOC), person name(G_PER), organization name(G_ORG), company name(G_COM)	1800	300	300

$$s(i,j) = \sum_{i-1}^{n} \left(W_{i-1,i} + P_{i,j} \right) \tag{4}$$

where W_{k*k} is a transformation matrix obtained through training. $s(i,j)$ represents the transition score of the j-th NER tag of the i-th token. For each token in the sentence, the NER tag with the highest transition score is selected as its corresponding NER label. Ultimately, we obtain a label sequence for the sentence denoted by $Y = \{y_1, y_2, \ldots, y_n\}$. Table 3 presents the example recognition results.

After acquiring all the NER labels for the tokens in SGs, the program scripts select the tokens with corresponding NER labels based on the integration database table fields and document order. The selected tokens are subsequently stored in a JSON file with their corresponding database table field name for further processing.

Data integration

The data integration stage imports the data after it is extracted to JSON format. This stage involves two main phases: firstly, removing redundant information from the data by aligning entities, and secondly, writing SQL code and corresponding read-write scripts to import the aligned JSON data. The latter is straightforward and abbreviated. We discussed the former phase in 3 steps.

1) **Word2Vec Training**. We trained a Word2Vec model through CAIL2018 [26] dataset to obtain word vectors specific to the legal field. The dataset in question comprises 5,730,302 legal documents sourced from China Judgments Online, encompassing a variety of document types, including judgments, verdicts, conciliation statements, decision letters, and notices. To explicitly train an arbitration-related Word2Vec model, a subset of the dataset containing only verdicts and conciliation statements was selected and preprocessed through operations such as tokenization. The objective of Word2Vec is to learn two matrices: a word embedding matrix $E \in R^{(V \times d)}$, where V is the vocabulary size and d is the dimensionality of the word embeddings. The training process can be divided into two phases:

In the first phase, each word in the corpus is transformed into a one-hot vector representation, denoted by x. For each word w_t in the corpus T, the input to a single-layer fully connected neural network is the one-hot vector x_t and the output is a probability distribution over the vocabulary, denoted by y_t. In the second phase, the neural network is trained to predict the context words c_t surrounding the input word w_t. The probability of predicting the context word c_t given the input word w_t is computed using the softmax function:

$$P(c_t|w_t) = softmax\left(y_t \cdot C\right) \tag{5}$$

where "·" denotes the dot product of two vectors. The objective function of the training process is to maximize the average log-likelihood of predicting the context words given the input words:

Table 3 NER extraction example

Extraction Example
The applicant, Zhang San [G_NAM], submitted an arbitration application on August 26th, 2021. The applicant is 36 years old [A_AGE], of Han [A_NAT], with an ID card number of 12345xxxx567 [A_ID], and male [A_GEN]. The applicant currently resides in Hunnan District, Shenyang City, Liaoning Province [A_OCC]
Sales contract dispute [C_CAS] with the case number of Fushun Arbitration Committee 2020 No. 032 [C_CODE]. The court session was held on June 8th, 2020 [G_DATE] at the Fushun Arbitration Committee [G_ORG]. Attendees included arbitrator Li Si [C_ARB], secretary Wang Wu [C_SEC], and clerk *Zhao Liu* [C_CLE]
The applicant provided evidence: 7 screenshots of WeChat chat records [C_EVI], which prove that the respondent did not fulfill the contract [C_CLE]
According to Article 22, Article 31 of the Arbitration Law of the People's Republic of China [C_LAW], Fuwa Heavy Industry Machinery Co., Ltd. [G_COM], to return 87,000.00 yuan (eighty-seven thousand) to the applicant, Jicheng Electric Manufacturing Co., Ltd. [G_COM], within 7 days [G_DATE]

The underline indicates the identified key information

$$L = (1/T) \sum_{t=1}^{T} \sum_{c \in C_t} logP(c|w_t) \quad (6)$$

As a result of this process, we obtained a word vector dictionary with n words $E = \{v_1, v_2, \ldots, v_n\}$ such that any arbitrary word w_i can be mapped to its corresponding word vector v_i.

2) **Entity Alignment**. This step addresses the issue of data redundancy resulting from multiple descriptions of the same object in data fields such as locations, company names, place names, and case types. For instance, to resolve redundancy in the *C_CAS* entity, expressions like "*contract issues*", "*contract problem*", and "*contract disagreement*" are replaced with "*contract dispute*" to standardize the descriptions. Entities that require alignment were traversed, and a dictionary was created to store pairs of entity word vectors. During entity traversal, the similarity between each entity and other entities within the dictionary is calculated based on their respective word vectors. Entities are grouped if their similarity exceeds a predefined threshold.
3) **Entity Integration**. For each entity group, the program script selects the data item that appears the most frequently and uses it to replace the values of other data items. Once all data items have been standardized in this manner, the corresponding JSON file is updated accordingly.

AI-enabled privacy protection

For data sharing and document disclosure in arbitration cases, it is necessary to filter the privacy of both the extracted arbitration database and arbitration award. To avoid the potential risk of privacy breaches arising from cloud services, we implement a privacy filtering mechanism to remove sensitive data related to privacy from the dataset before uploading it to cloud storage. Within the arbitration application context, the cloud services primarily consist of generic services that do not require sensitive data, such as support for knowledge graphs, SMS services, and other similar offerings. This section discusses our AI-enabled privacy filtering method for each object separately to address the challenge.

Database privacy protection

We employ an anonymous privacy protection algorithm based on *t*-Closeness to filter sensitive attribute columns in the database. The respective attribute columns are removed for Identifier Attributes (IAs), such as identification or phone numbers. Meanwhile, for Quasi-Identifier (QI) and Sensitive Attribute (SA) anonymization, we follow the specific steps of the t-Closeness algorithm, which are as follows:

1) **Identify the QI and SA attributes.** We determine each table's IA, QI, and SA attributes and discard the IA attribute column. We then obtain the filtered attribute set $QI_s = \{QI_1, QI_2, \ldots, QI_n\}$. Take the *PARTICIPANT_INFO* table as an example; its IA attributes are *participant_id* and *identity_code*, its QI attributes include *name*, *gender*, *age*, and *department*, and its SA attribute is the *case*.
2) **Construct equivalence classes.** We initialize empty equivalence class D and iteratively add the top k records in the data table to D. For each iteration, we calculate EMD (Earth Move Distance) [27] to determine whether the distribution difference is less than the threshold; otherwise, we move to the next iteration. This iteration runs until it is no longer possible to allocate a record into any equivalence class or no more records are left.
3) **Generalization.** We generalize each equivalence class uniformly to obtain anonymized data, that is, anonymize privacy attributes in each equivalence class using a specific privacy anonymization strategy.

As shown in Table 4, sub-table (a) on the left randomly selected eight rows of the *PARTICIPANT_INFO* table, where the IA attribute is the *id* field, $QIs = \{age, zipcode\}$, and SA is the *case* field; After discarding the IA attribute and anonymizing t-Closeness, the right

Table 4 t-Closeness anonymous example

No	Id	Age	Zip code	Gender	Case
(a) Original data of the *PARTICIPANT_INFO* table					
1	123	25	110000	male	contract dispute
2	124	29	113000	female	property dispute
3	125	35	118000	female	property dispute
4	126	36	122000	male	labor dispute
5	127	43	124000	male	contract dispute
6	128	30	113000	female	contract dispute
7	129	27	115000	male	labor dispute
8	130	55	125000	male	labor dispute
(b) t-Closeness anonymous data (t = 0.25, k = 2)					
1	Null	2*	11*	male	contract dispute
2	Null	2*	11*	female	property dispute
7	Null	2*	11*	female	labor dispute
3	Null	[30, 37]	1*	male	property dispute
4	Null	[30, 37]	1*	male	labor dispute
6	Null	[30, 37]	1*	female	contract dispute
5	Null	> = 38	12*	male	contract dispute
8	Null	> = 38	12*	male	labor dispute

sub-table (b) is obtained, which satisfies $t = 0.25$ and 2-diverse.

The privacy filtering process described above, implemented through an automated script, effectively addresses the two issues associated with manual privacy filtering mentioned earlier. On the one hand, the *t*-Closeness-based privacy filtering method considers the data distribution more rigorously. On the other hand, using an automated de-identification script eliminates the need for human involvement, thereby reducing the risk of privacy breaches.

Text field privacy protection

While privacy filtering schemes based on anonymity technology effectively address privacy concerns when the database fields themselves consider private information in their entirety, they prove insufficient in the current application scenario because long text fields within the database may contain privacy-sensitive information. Such fields include the '*description*' field in the *APPLICATION_ATTACH* table, the *argue_info* field in the *COURT_RECORD* table, and the demand for the publication of the entire arbitration award. To achieve comprehensive privacy protection, we augment the previously described method with NER techniques from NLP, explicitly targeting the filtering of long texts containing sensitive information.

We continue to employ the NER model BERT-BiLSTM-IDCNN-CRF in the information extraction to identify entities within the text that require privacy filtering. Firstly, we select entity types that pertain to sensitive information, such as location (*G_LOC*), name (G_PER), organization name (*G_ORG*), and personal information entities such as gender (*A_GEN*), age (*A_AGE*), occupation (*A_OCC*), and nationality (*A_NAT*) and assign specific filtering rules to each entity. Some of the text is replaced with '*', such as replacing *'Zhang Sanbao'* with '*Zhang* *', '26' with '2*', and '*Fushun Fertilizer Company*' with '*xx Company*'. Secondly, we apply the trained NER model to relevant long text fields to recognize named entities and filter them according to the rules. Table 5 shows the replacement examples. The privacy filtering method for arbitration documents is similar to database filtering. The difference lies in identifying personal names, where name (*G_PER*) entities, including arbitrators and agents, are not filtered.

Complexity analysis

The privacy-preserving part includes two algorithmic components: equivalence class construction in t-closeness and NER model, which will be analyzed separately below.

1) **Complexity of equivalence class construction**. Equivalence class construction partitions a set of n elements into n/k sets, each containing k elements, where k is the anonymization parameter. The time complexity of constructing an equivalence class is the product of the number of available options for selecting each element, which can be calculated as $O\left(\prod_{j=0}^{k-1} n - j\right)$. number of elements in the set. By summing up the time complexities of constructing all equivalence classes, the time complexity of the entire equivalence class construction process can be determined. This can be expressed as $O\left(\sum_{i=0}^{n/k} \prod_{j=0}^{k-1} (n - j - (i-1) * k)\right)$, where i represents the i-th constructed set and j represents the j-th selected element. This time complexity grows exponentially with the size of the set and parameter k, and can be approximated as $O\left(n^k\right)$.
2) **Complexity for NER**. The time complexity of NER is largely determined by the forward propagation computation of the BERT model in the BERT-BiLSTM-IDCNN-CRF model. Specifically, assuming an input sequence length of n, a word embedding dimension of V, a BERT hidden layer dimension

Table 5 Table column filter example

Table Column	Filtered Example
APPLICATION_ATTACH.desctiption	The respondent (Zhang San [G_NAM] -> Zhang*) borrowed 150,000 yuan from us on (August 15th, 2021 [G_DATE] -> "x year x month x day") (Fushun Fertilizer Company [G_ORG] -> "* company")), is aware of this matter but has not taken any action
COURT_RECORD.argue_info	The main point of dispute between the two parties is whether the oral agreement on interest between the applicant (Zhang San [G_NAM] -> Zhang*) and the respondent (Li Si [G_NAM] -> Li*) is valid, as well as whether (Fushun Fertilizer Company [G_ORG] -> "* company") belongs to has joint liability
REPLY_BRIEF.confirmed	We acknowledge that on (November 15th, 2021 [G_DATE] -> "x year x month x day"), (Zhang San [G_NAM] -> Zhang *) promised a interest rate of 5%
EVIDENCE_RECORD.description	This evidence is the loan agreement signed by (Zhang San [G_NAM] -> Zhang*) and the respondent (Li Si [G_NAM] -> Li*) on (November 15th, 2021 [G_DATE] -> "x year x month x day" in (Heping District [G_LOC]- > xxxx) of (Shenyang [G_LOC]->xxxx)

of H, and a BERT layer count of L, the BERT forward computation complexity can be expressed as the sum of the complexities of the embedding, self-attention, and forward layers, which is $O(((V+n+2)*H_1)+(12L_1*H_1{}^2)+(8L_1H_1^2+5L_1H_1))$. In our practice, we selected BERT-base as the training model with H = 256 and L = 12. This model has a total of 110 M parameters, and its FLOPs (floating point operations) is approximately $1.0*10^{11}$.

Functional evaluation

This section presents the evaluation results from the perspective of Functional Evaluation, where we selected commonly used evaluation metrics to assess the performance of both the AI model and the anonymity-based privacy filtering.

Setup

The BERT classification model dataset is derived from OCR-identified Poorly-Formatted Data. As shown in Table 6, we define 4–8 classification tags for each document type based on its content. After then, we selected 600 Well-Formatted Data and 800 Poorly-Formatted Data and labeled them accordingly. Among them, 80 percent is for training, 10 percent is for testing, and 10 percent is for validation.

As shown in Table 6, our experiments were conducted on the server. The classification model we selected is BERT-based, Chinese, with a hidden layer dimension of 768, 12 attention heads, and two fully connected feed-forward layers with dimensions of 3702 and 768, respectively. The fine-tuning process uses the Adam optimizer, with a learning rate set to 5×10^{-5}, a batch size of 16, and a sequence length of 256. Sentences exceeding the maximum length are truncated.

We set the model parameters to their default values as specified in the original paper. These values included a transformer layer number of 12, a hidden layer size of 768, an attention layer number of 12, an LSTM dimension of 64, a learning rate of 0.01, a dropout rate of 0.1, a clip of 5, an optimizer of Adam. However, we adjusted the batch size to 16 to better suit our specific training environment.

The experiments run on Core i7-13700KF CPU, GeForce RTX 2080Ti GPU,16G RAM, and Ubuntu 18.04 operational system.

AI models evaluation

The evaluated models are the BERT model for dividing document semantic paragraphs into groups, and the BERT-BiLSTM-IDCNN-CRF NER model to extract key entities. We adopt precision, recall, and F1-score metrics as our model metric. For the BERT classification model, we calculate the corresponding metrics for each class and then draw the mean values. For the NER model, we calculate the metrics for each named entity type individually and then draw the mean values.

As shown in Table 7, we compared two schemes for the classification task: training separate classification models for each data type, and employing a single classification model for all data types. Both models show a competitive classification accuracy. While the single model scheme did underperform the multiple model scheme in terms of classification accuracy, we ultimately selected this approach due to its relatively lower construction and training costs.

Our trained model achieved 85% performance on all three metrics regarding the NER model. There is a roughly 5% discrepancy between our model performance indicators and those reported in the original paper, i.e., precision = 86.16%, recall = 78.99%, F1-score = 84.54%. This difference may be attributed to the small semantic gap between custom annotation entities, insufficient training data, or the variable quality of the source data. Nevertheless, the model remains capable of meeting our information extraction requirements.

Table 7 Model evaluation result

Model	Type No	P(%)	R(%)	F1(%)
Classification(type)	1	94.17	93.45	93.80
	2	89.76	82.47	85.96
	3	96.32	93.34	94.81
	4	83.44	84.01	83.72
	avg	**90.92**	**88.31**	**89.57**
Classfication(one)		**89.21**	**83.42**	**86.22**
NER		**82.54**	**78.24**	**80.33**

Table 6 Classification label set

No	Data type	Labels
1	Arbitration Application Form	applicant_sec, request_ sec, case_ sec, appendix_sec
2	Court Record	informatiob_ sec,, dispute_sec confrontation_sec, evidence_sec
3	Evidence	information_sec, detail_sec, verified_sec, appendix_sec
4	Arbitration Award	start_sec, participant_sec, process_sec, case_sec, opinion_sec, law_sec

Privacy protection evaluation

The evaluation of privacy filtering methods can be designed from two perspectives: firstly, the effectiveness of privacy protection, which pertains to the extent to which the method provides privacy protection. Secondly, the amount of information retained after privacy filtering. While the effectiveness of privacy protection is difficult to quantify, the *t*-Closeness algorithm can ensure the effectiveness of privacy filtering. Compared to crude manual filtering methods, the *t*-Closeness algorithm can effectively prevent privacy attacks such as homogeneity attacks, background attacks, and similarity attacks. Our primary focus is on conducting a quantitative analysis of the amount of information lost after privacy filtering and examining the amount of information lost under varying degrees of privacy filtering.

Using the *PARTICIPANT_INFO* table with 6000 records as the representative evaluation object, we selected the Discernibility Metric Cost (DMC) [28], which computes the number of records that are indistinguishable from each other, and Minimal Average Group Size (MinA) as measurement metrics. We perform *t*-Closeness with different parameters (*k*,*l*) to conduct privacy filtering on the data items in the *PARTICIPANT_INFO* table. Subsequently, we computed the DMC and MinA metrics on the privacy filtering results, and the outcomes are displayed in Fig. 9.

First, the DMC and MinA values increase with k and l. s, the values under $t=0.20$ are higher than those under $t=0.25$. The experimental results indicate that data quality continuously deteriorates with increased data anonymization extent. Therefore, it is essential to balance the de-identification degree and data quality when practicing privacy filtering. In our experience, parameters $k=4$, $l=4$, and $t=0.20$ essentially met our privacy filtering requirements.

Human-AI comparison

In this section, we analyzed the pros and cons of the AI-enabled LDI compared with the manual LDI. This analysis complements the functional evaluation discussed previously and fully demonstrates the effectiveness of our method. First, we assume manual results are ground truth and list the errors of AI-enabled LDI. Second, we compare the two methods in qualitative and quantitative manners. The experimental results demonstrate that our proposed method achieves favorable outcomes in terms of both accuracy and time consumption.

Setup

Two competitors

To compare the pros and cons of the two LDI methods, we formed Manual and AI teams to perform LDI with the two methods, respectively.

- **Manual LDI team**. The team consisted of an arbitration expert, a system administrator for Cloud Arbitration Court, and four internships. The former two are responsible for formulating integration rules, and the latter is responsible for performing integration tasks.
- **The AI team**. The team consisted of two internships for training the model, writing scripts, and collecting program output.

We monitored the entire integration process and effectiveness of the two teams as the basis for subsequent comparative analysis.

Correctness of AI results comparing manual results

Let the manual result be the ground truth, we defined the *correct*(*table*, *row*, *column*) to calculate whether the AI result of the given *table*, *row*, and *column* is correct. The *correct*() is a binary value, representing whether the AI result is consistent with the manual one. For key columns such as *name*, *age*, and *identity_code*, *correct*() check their equivalence. For text columns such as *description*, *opinion*, and *argue_info*, *correct*() check their textual similarity (shown in Fig. 10), which is within the (0,1] range. If the similarity is larger than a threshold ε, then *correct*() return 1.

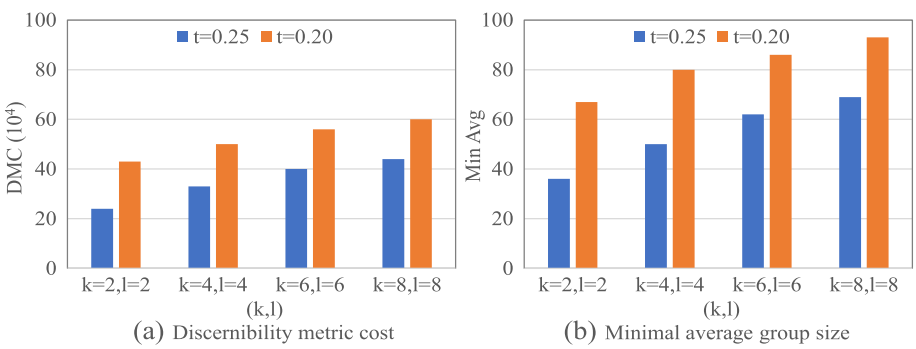


Fig. 9 Quality measures of privacy-filtered data

Accuracy of AI-enabled LDI

Based on the *correct*(), we define the accuracy of the AI-enabled LDI as follows:

$$Acc = \frac{\sum_i^{table}\sum_j^{row}\sum_l^{column} correct(i,j,k)}{\sum_i^{table}\sum_j^{row}\sum_l^{column} 1} \quad (7)$$

where *Acc* is the short name of accuracy; *table*, *row*, and *column* represent the number of database tables, rows, and columns, respectively.

Recognition errors

Comparing the integration results of AI-enabled LDI and manual LDI, we find two types of errors that commonly arise during the AI-enabled process:

- **Failed to extract target information.** This type of error typically implies a missing attribute of the integrated data item. For example, name information sometimes appears in a handwritten form, resulting in a challenge for the AI-enabled method to recognize and extract. Consequently, the "*name*" field in the *PARTICIPANT_INFO* table may be empty.
- **Integrating wrong data.** This type of error occurs more frequently than the previous one and typically implies setting table fields to inaccurate values. For the example of court records, model semantic understanding bias results in the erroneous identification of the applicant as the arbitration *agent*, thereby importing applicant information records into the *ATTORNEY_INFO* table.

Recognition errors occur in various attributes; however, those containing larger textual data are potential occurrences. For example, the *addresses* attribute that shows the applicant's residence address, the integration of company addresses, or contact information into the applicant's information, is likely to integrate wrong data.

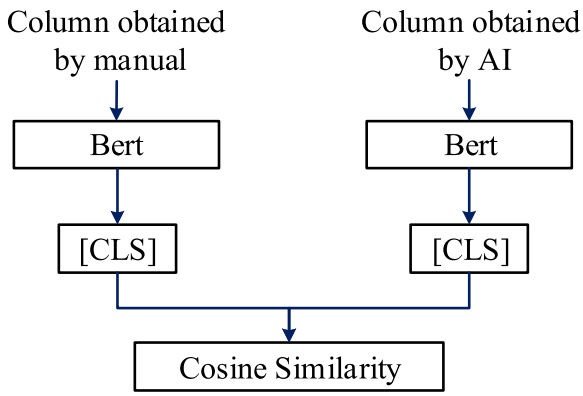


Fig. 10 Textual similarity calculation

The recognition errors primarily stem from two reasons. First, the diversity of source data formats and writing styles increases the difficulty of OCR recognition and the semantic understanding of AI models. For example, the writing styles of different individuals vary, bringing the challenge to the model generalization. Second, regular expressions cannot adapt to changes in text structure since they can only match predefined patterns. Third, the semantic understanding and information extraction capabilities of AI models are limited. For example, the BERT model has limited abilities in Chinese semantic understanding, resulting that the model fails to comprehend complex source text data.

Although the above problems are not completely solved in the current method, we have minimized the occurrence frequency of the problems by combining structured extraction methods and multiple AI models, and a quantitative comparison analysis is carried out in the next section to verify our effect.

Qualitative comparison

Under the comparative setting described earlier, we selected integration error rate, integration consumption, and process management difficulty as the comparative factors to compare the two methods qualitatively.

As shown in Table 8, the AI-enabled method has certain advantages over the manual extraction method in terms of labor cost, and accuracy due to its features, such as being AI-based, automated, and having fixed extraction patterns. However, limited by the semantic understanding ability and poor interpretability of end-to-end AI models, the AI-enabled method has higher error rates. It is relatively difficult to locate and resolve errors. Although the manual extraction method is superior in terms of extraction effectiveness and flexibility, it involves the participation of more staff with different roles, making the process and quality management more challenging and requiring more labor.

In terms of the privacy filtering effect, manual privacy filtering adopts a fixed privacy field filtering method for manual filtering, which lacks consideration of the entire data and results in a large amount of information loss, reducing data availability. Our AI-enabled LDI implements privacy filtering based on *t*-closeness and NER recognition, which fully considers the overall data distribution and maximizes the retention of data information.

Quantitative comparision

To evaluate the effectiveness of our AI-enabled LDI method, we randomly selected 2000 historical paper

Table 8 Qualitative comparison

Aspects	AI	Manual
Error Rate		
Pro.	The error rate is relatively stable and does not vary with changes in workload	Integration errors are relatively fewer and smaller
Con.	Limited by AI model ability, there is a tendency for more integration errors to occur.	As the workload increases, the probability of integration errors occurring also increases.
Integration Consistency		
Pro.	Fixed AI model and program ensure consistency in the effectiveness of integration.	It is more flexible and facilitates rapid adaptation to new integration rules.
Con.	Lack of flexibility makes it difficult to respond to changes in integration requirements.	Different understandings of integration rules among staff members lead to poor consistency.
Cost		
Pro.	Overall, it saves a significant amount of labor and time.	No outside staff participation is required.
Con.	Additional computer experts are needed to design and write relevant programs.	More labor cost and time consumption.
Difficulty of quality management		
Pro.	Locating bugs from program output and logs is relatively simple	By communicating with relative staff, the cause of errors and solutions can be quickly determined
Con.	Lack of interpretability of the AI model leads to integration results that cannot be explained, and errors cannot be tracked.	Locating errors requires interaction with humans, which is more complex and less predictable.

document samples as our test data, obtained both integrated data of manual LDI and our AI-enabled LDI method, and evaluated the accuracy *Acc* as well as the time consumption.

Accuracy

Table 9 presents the differences between the AI-enabled LDI and manual LDI from three perspectives: *data table*, *source data*, and *overall effects* under different thresholds ε:

- **Data table perspective**

  This perspective primarily focuses on analyzing data integration accuracy (*Acc*) in the data tables. For example, the *APPLICATION* table and *PARTICIPANT_INFO* are extracted from documents of the *Arbitration Application Form*. The *Acc* is as high as about 0.83 since these documents have a clear format and are relatively short. On the contrary, the *COURT_RECORD* table and *ARBITRATION_AWARD* table have heavy demands on the semantic understanding ability. The accuracy decreases to 0.60 ~ 0.73 because the tables have a more complex data format, and the data are extracted from longer paragraphs.
- **Source data perspective**

  Well-Formatted Data has *Acc* between 0.85 and 0.93, indicating that the AI-enable LDI effectively solves the integration problem on such source data. In contrast, Poorly-Formatted Data only reaches *Acc* from 0.61 to 0.78, indicating that the extraction accuracy on such source data still has space to improve.
- **Overall perspective**

  The results in Table 9 demonstrate that AI-enabled LDI can achieve an overall recognition accuracy of

Table 9 AI-enabled LDI ACC

	Dimension	Integrated Rows	Integrated Columns	Acc (ε = 0.4)	Acc (ε = 0.6)	Acc (ε = 0.8)
Data table	APPLICATION	2000	7	0.90	0.85	0.76
	PARTICIPANT_INFO	3623	10	0.83	0.83	0.83
	COURT_RECORD	2835	6	0.73	0.68	0.60
	ARBITRATION_AWARD	2000	6	0.76	0.70	0.65
Source Data	Well-Formatted Data			0.93	0.87	0.85
	Poorly-Formatted Data			0.78	0.72	0.61
Overall				**0.80**	**0.74**	**0.65**

0.67 ~ 0.80. As the threshold increases, the recognition accuracy decreases, with the lowest accuracy of 0.67 occurring when $\varepsilon = 0.4$.

Time consumption

Effectiveness is the advance of AI-enabled LDI. The same as accuracy comparison, we record the time consumption for each step, namely the preparation stage for defining integration rules, and the data integration stage for executing LDI. The time consumption for both methods was measured in 'person-hours', as shown in Table 10.

The AI-enabled LDI required additional work during the preparation stage due to the need to write and train relevant models. However, during the data integration stage, the AI-enabled LDI saved a significant amount of the workforce with automatic and programmatic integration, resulting in a 92% reduction in time consumption. Offset by prepare stage, the AI-enabled LDI could reduce 59% of overall time. Such advantages become more prominent as the legacy data volume increases. In our practical integration work for the Cloud Arbitration Court in Liaoning Province, we have achieved a time-saving of 90% using our AI-enabled LDI method, according to the historical integration experience.

Conclusions and future work

This paper proposes an AI-enabled LDI method for the Regional Cloud Arbitration Court, which ensures privacy protection through filtering techniques while integrating data. Firstly, we study the content and format features of historical law-related documents and implement a conversion from source document data to database data based on structured and unstructured extraction methods using NLP techniques. Secondly, we utilize anonymization techniques and NLP methods to filter sensitive data and achieve privacy protection. Experimental results demonstrate that our approach achieves similar extraction results to a manual extraction and significantly reduces labor costs during the information integration stage, effectively advancing the data integration process of local arbitration.

However, our proposed method still faces some limitations. Firstly, it is only suitable for extracting information from pure text data, and its ability to process more complex unstructured text data such as tables and images is limited.

Table 10 Time consumption comparison

Integration Type	Stage1: Prepare	Stage2: Integration	Total
Manual	64	280	344
AI-enabled	120	20	140
Saved	-56	260	204
Saved Rate	**-84%**	**92%**	**59%**

Secondly, the AI models used in our method are highly sensitive to the quality of training data and may not perform well in scenarios where data is scarce. For future work, we plan to continue improving the limitations of our current work. Firstly, we will further investigate the extraction of unstructured document information with rich visual characteristics. Secondly, we will explore the direction of end-to-end privacy filtering to address the complexity and lack of transferability of the current privacy filtering solution.

Abbreviations

AI	Artificial Intelligence
LDI	Legacy Data Integration
BIM-GIS	Building Information Modeling, Geographic Information System
OCR	Optical Character Recognition
NER	Named Entity Recognition
BERT	Bidirectional Encoder Representation from Transformers
NLP	Natural Language Processing
PPDP	Privacy-Preserving Data Publishing
BIO	Begin, Interior, and End

Acknowledgements

The authors would like to thank all anonymous reviewers for their invaluable comments.

Authors' contributions

Jie Song contributed to the requirement, idea, technical solution, and Sections 1, 3, and 5. HaiFei Fu contributed to the implementation and Sections 4, 6, and 7. Tianzhe Jiao contributed to Sections 2 and 8. Dongqi Wang is the corresponding author and contributed to the funding and proofreading. All authors have read and approved the manuscript.

Funding

This paper is supported by the Fundamental Research Funds for the Central Universities (No. N2217002); and the Natural Science Foundation of Liaoning Provincial Department of Science and Technology (No.2022-KF-11-04).

Availability of data and materials

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

Declarations

Competing interests

The authors declare no competing interests.

Received: 15 June 2023 Accepted: 5 August 2023
Published online: 14 October 2023

References

1. Rashid A, Chaturvedi A (2019) Cloud computing characteristics and services: a brief review. Int J Comput Sci Eng 7(2):421–426
2. Zheng GG (2020) China's grand design of people's smart courts. Asian J Law Soc 7(3):561–582. https://doi.org/10.1017/als.2020.20
3. Anatoly Tikhanovich K, Alexander Vladimirovich S, VeronikaAleksandrovna M (2021) On the effectiveness of the digital legal proceedings model in Russia. Mathematics 9(2):125. https://doi.org/10.3390/math9020125
4. Suhanto A, Hidayanto AN, Naisuty M, Bowo WA, Ayuning Budi NF, Phusavat K (2019) Hybrid cloud data integration critical success factors: a case study at PT Pos Indonesia. In: 2019 Fourth International Conference on Informatics and Computing (ICIC). pp 1–6. https://doi.org/10.1109/ICIC47613.2019.8985748

5. Zhou X, Hu Y, Wu J, Liang W, Ma J, Jin Q (2022) Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in industrial IoT. IEEE Trans Industr Inf. https://doi.org/10.1109/TII.2022.3170149
6. Jia Y, Liu B, Dou W, Xiaolong Xu, Zhou X, Qi L, Yan Z (2022) CroApp: a CNN-based resource optimization approach in edge computing environment. IEEE Trans Industr Inf 18(9):6300–6307
7. Zhou X, Xu X, Liang W, Zeng Z, Yan Z (2021) Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT. IEEE Internet Things J 8(16):12588–12596. https://doi.org/10.1109/JIOT.2021.3077449
8. Dhayne H, Haque R, Kilany R, Taher Y (2019) In search of big medical data integration solutions - a comprehensive survey. IEEE Access 7:91265–91290. https://doi.org/10.1109/ACCESS.2019.2927491
9. Leng S, Lin J-R, Li S-W, Hu Z-Z (2021) A data integration and simplification framework for improving site planning and building design. IEEE Access 9:148845–148861. https://doi.org/10.1109/ACCESS.2021.3124010
10. Reda R, Piccinini F, Martinelli G, Carbonaro A (2022) Heterogeneous self-tracked health and fitness data integration and sharing according to a linked open data approach. Computing 104(4):835–857. https://doi.org/10.1007/s00607-021-00988-w
11. Habib K, Saad MHM, Hussain A, Sarker MR, Alaghbari KA (2022) An aggregated data integration approach to the web and cloud platforms through a modular REST-based OPC UA middleware. Sensors 22(5):1952. https://doi.org/10.3390/s22051952
12. Prasath N, Sreemathy J (2021) A new approach for cloud data migration technique using talend ETL tool. In: 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). pp 1674–1678. https://doi.org/10.1109/ICACCS51430.2021.9441898
13. Rodriguez A, Chen Y-L, Argueta C (2022) FADOHS: framework for detection and integration of unstructured data of hate speech on Facebook using sentiment and emotion analysis. IEEE Access 10:22400–22419. https://doi.org/10.1109/ACCESS.W2022.3151098
14. Liu J, Abeysinghe R, Zheng F, Cui L (2019) Pattern-based extraction of disease drug combination knowledge from biomedical literature. In:2019 IEEE International Conference on Healthcare Informatics (ICHI). pp 1–7. https://doi.org/10.1109/ICHI.2019.8904473
15. Nguyen M-T, Le DT, Le L (2021) Transformers-based information extraction with limited data for domain-specific business documents. Eng Appl Artif Intell 97:104100. https://doi.org/10.1016/j.engappai.2020.104100
16. Kerroumi M, Sayem O, Shabou A (2021) VisualWordGrid: information extraction from scanned documents using a multimodal approach. In: Barney Smith EH, Pal U (eds) Document analysis and recognition – ICDAR 2021 workshops. Springer International Publishing, Cham, pp 389–402
17. Liu S, Ma J, Feng X (2019) Transparent access and integration of heterogeneous encrypted database in hybrid cloud environment. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). pp 1–6. https://doi.org/10.1109/ICC.2019.8761975
18. Alqarni AA (2021) A secure approach for data integration in cloud using Paillier homomorphic encryption. J Basic Appl Sci 5(2):15–21
19. Ren W, Ghazinour K, Lian X (2022) kt-Safety: graph release via k-Anonymity and t-Closeness. IEEE Trans Knowl Data Eng 1–12. https://doi.org/10.1109/TKDE.2022.3221333
20. Khan P, Khan Y, Kumar S (2021) Single identity clustering-based data anonymization in healthcare. In: Bansal JC, Paprzycki M, Bianchini M, Das S (eds) Computationally intelligent systems and their applications. Springer Singapore, Singapore, pp 1–9. https://doi.org/10.1007/978-981-16-0407-2_1
21. Iwendi C, Moqurrab SA, Anjum A, Khan S, Mohan S, Srivastava G (2020) N-sanitization: a semantic privacy-preserving framework for unstructured medical datasets. Comput Commun 161:160–171. https://doi.org/10.1016/j.comcom.2020.07.032
22. Moqurrab SA, Anjum A, Khan A, Ahmed M, Ahmad A, Jeon G (2021) Deep-confidentiality: an IoT-enabled privacy-preserving framework for unstructured big biomedical data. ACM Trans Internet Technol 22(2):1–21. https://doi.org/10.1145/3421509
23. Li Z, Xiaolong Xu, Hang T, Xiang H, Cui Y, Qi L, Zhou X (2022) A knowledge-driven anomaly detection framework for social production system. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/TCSS.2022.3217790
24. Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805. Available: http://arxiv.org/abs/1810.04805
25. Chang Y, Kong L, Jia K, Meng Q (2021) Chinese named entity recognition method based on BERT. In:2021 IEEE International Conference on Data Science and Computer Application (ICDSCA). pp 294–299. https://doi.org/10.1109/ICDSCA53499.2021.9650256
26. Xiao, et al. (2018) CAIL2018: a large-scale legal dataset for judgment prediction. CoRR abs/1807.02478. Available: http://arxiv.org/abs/1807.02478
27. Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. Int J Comput Vision 40(2):99
28. Bayardo RJ, Agrawal R (2005) Data privacy through optimal k-anonymization. In: 21st International Conference on Data Engineering (ICDE'05). pp 217–228. https://doi.org/10.1109/ICDE.2005.42

Publisher's Note