Research on the Construction Technology of University Data Resource Catalogs Based on Machine Learning

Ying Zhang[1,2,a], Ying Guo[1,2,b*], Shangxu Liu[1,2,c], Xiaohan Yang[1,2,d], Bowen Sun[1,2,e]

[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2]Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan, China

[a]zhangyingdata@163.com,[b*]guoying@sdas.org
[c]19862177680@163.com,[d]y_angxiaohan@163.com,
[e]1723713231@qq.com

Abstract. Currently, in the process of information system construction in universities, the diverse construction of various departmental business information systems leads to issues such as the diversification of data characteristics and the phenomenon of information silos. This paper aims to construct a unified data resource catalog for universities and conducts in-depth research. Under the current national strategy of digitalization in education and the requirements for informatization development in universities, establishing a clear and orderly data resource catalog is crucial. It helps in building a comprehensive digital architecture, enhancing the utilization of data value, and supporting data sharing and decision-making. Traditional data integration faces challenges such as interference between integration tools and business systems, inability to synchronize metadata in real-time, inconsistency in data standards among different business systems, and lack of metadata semantic information. To address these issues, this paper proposes a method for university data resource catalog based on the Hudi Lakehouse, and details key works in four aspects, including data lake research, the design of university data mapping dictionaries, column semantic recognition methods, and data resource catalog construction technology. It effectively overcomes problems such as connection interference, metadata change perception, and metadata column semantic information recognition, establishing a unified data resource catalog for universities. This achievement is expected to provide strong support for university data management and governance, promote data sharing and utilization, and have a positive reference significance for the future operational models and informatization construction of universities.

Keywords: Data Catalog, Big Data, Data Governance, Natural Language Processing.

M. Yu et al. (eds.), *Proceedings of the 2024 5th International Conference on Big Data and Informatization Education (ICBDIE 2024)*, Advances in Intelligent Systems Research 182,
https://doi.org/10.2991/978-94-6463-417-4_3

1 Introduction

In today's data-driven era, universities have constructed various departmental business information systems in the process of informatization, accumulating a large amount of data. Due to the diversity of data in terms of type, structure, and source in these systems, it's difficult for data users to cross departments and find needed data in a one-stop manner. Additionally, university leadership cannot fully grasp the overall data assets, leading to a severe "information silo" phenomenon between departments. This hinders the full utilization of data value and the integrated development of university informatization. In line with implementing the national digital education strategy and the requirements for the intelligent and informatized construction of universities, effective organization and building of a clear, orderly unified university data resource directory[1] can help construct a comprehensive digital architecture for universities. It enables leadership and management personnel to comprehensively understand university data assets and assists departmental data users in quickly and centrally locating the data they need. This improves data insight and analysis, supports data governance[2], provides efficient and convenient search conditions for data sharing, and supports precise management and decision-making with data. However, current research and application of building data directories are mostly concentrated in government, medical, and corporate fields. The construction methods are often simple collations of existing business system metadata and lack attention to university-specific data directory construction technology. Additionally, most researchers have not addressed the practical problems that universities urgently need to solve in metadata collation.

First, traditional data integration ETL tools can disrupt connections to the original relational databases of business systems during data extraction. Business system managers often prefer not to grant ETL tools full access to the data, typically offering limited access through data views. This approach may obscure some relationships between data, thereby impacting the quality of the data resource directory and the data resource platform. Second, the synchronization of data from business systems to the data resource center typically happens on a scheduled or real-time basis. However, changes in metadata due to system upgrades, data updates, or changes in the original business system requirements are not promptly reflected in the data resource directory. These changes often require the business system software vendor to perform later upgrades to data views or interfaces. Third, due to technical protection measures by business system software vendors, sufficient metadata is not injected into the database table structures. This results in a lack of semantic information in the metadata obtained by the data resource center through traditional ETL methods. Issues such as synonymous but differently named column projection metadata can arise. Current traditional machine learning techniques for column semantic identification rely solely on the attribute values of individual database columns and lack the capability to capture the contextual semantic information of relational data tables. This makes it difficult to accurately distinguish column semantics. While attention mechanisms can be used to obtain the weight of each word for capturing contextual semantic information, they focus mainly on local continuous word sequences. This approach only provides partial contextual semantic features of relational data tables[3], limiting the ability to capture the full scope of global semantic features of the tables. Additionally, traditional models are sensitive to the order of columns in relational data, which can significantly impact the accuracy of model predictions, making it challenging to support consistent categorization and uniform metadata descriptions for the data assets of multiple business systems in universities.

In response to the issues identified in the aforementioned technologies, this paper primarily investigates a method for creating a unified data resource directory for universities based on Hudi[4], an integrated lakehouse architecture. The research is broken down into stages and elaborates on the construction steps. The focus of this study is on four key areas: data lake entry research, university data mapping dictionary design, column semantic identification methods, and data resource directory construction technology. By integrating with the data governance platform project of Qilu University of Technology, this research effectively overcomes the existing technological shortcomings such as connection interference, inability to promptly sense changes in the original system data structure, inability to recognize the meaning of metadata[9],and lack of uniform data standards. It has established a clear and orderly unified data resource directory for universities. This directory now manages university data resources as data assets, facilitating maintenance, management, analysis, and mining. This approach aims to unlock the value of data, promote data sharing, and achieve the goal of university data governance. The findings and methods applied in this research are significant for the management of university data assets and can provide insights for future operational models in higher education institutions.

2 Relevant background and work

2.1 Lake-Warehouse Integration

Lake-Warehouse Integration is a data management system based on open formats, operating on low-cost storage and providing traditional analytical DBMS functionalities. It has now superseded traditional data lakes[10], effectively merging the advantages of both data lakes and data warehouses. It is built upon the cost-effective, open-format data storage architecture of data lakes, while also inheriting the high-performance, transactional data processing, and management capabilities of data warehouses. This integration enables the unified storage of various data types from multiple business systems. Lake-Warehouse systems include technologies like Delta Lake, Apache Hudi, and Apache Iceberg, facilitating seamless scheduling and management of data between lakes and warehouses. Data can be accessed, queried, and analyzed through a unified interface at a higher level, effectively resolving issues such as complex ETL logic, challenging metadata changes, and metadata inconsistencies.

2.2 Column Semantic Recognition

Column Semantic Recognition refers to the process of understanding the semantics of each column in a relational table, which includes methods based on knowledge bases, statistical features, and deep learning. Knowledge-base-based recognition involves matching columns in a text with a pre-built or existing external database to determine column semantics, as demonstrated by Chen J et al., who used knowledge bases for automatic annotation of columns in Web tables. Statistical feature-based recognition relies on statistics like word frequency, co-occurrence frequency, and context to infer the semantics of a column. Deep learning-based recognition employs neural networks to learn column semantics in textual data, often using models like CNN, RNN or Transformer[11] with self-attention mechanisms. Ding et al. proposed the CCA model, which concatenates all cell values of a relational data column into text and fine-tunes on the pre-trained model BERT, classifying and mapping to semantic labels.

3 system design

This paper primarily investigates a university data resource catalog method based on the integrated Hudi lakehouse concept and outlines the construction steps in stages. The research focuses on four main areas: data lake entry, design of university data mapping dictionaries, column semantic recognition methods, and data resource catalog construction technology. The process begins with extracting original data from various university business systems into the lakehouse and partitioning initial metadata. Following this, a preliminary version of the university data standard mapping dictionary is built based on the GB/T 29808 national standard. The Chinese abbreviations in the data standards are then standardized into model semantic category labels. Additionally, semantic recognition is performed on unlabeled metadata extracted into the lakehouse using the CSR model. This involves correcting annotations based on model semantic category labels and updating these back into the lakehouse's metadata annotations. The lakehouse's linked backfill of existing annotations and corrected metadata annotations are then supplemented into the standardized metadata descriptions and the data warehouse's university data standard mapping dictionary. Finally, using the university metadata standard mapping dictionary in the data warehouse, a classified and hierarchical university data asset catalog is constructed, enabling the publication and search of the data asset catalog.The system for constructing a university relational data resource directory based on Lake-Warehouse Integration is illustrated in Figure 1.

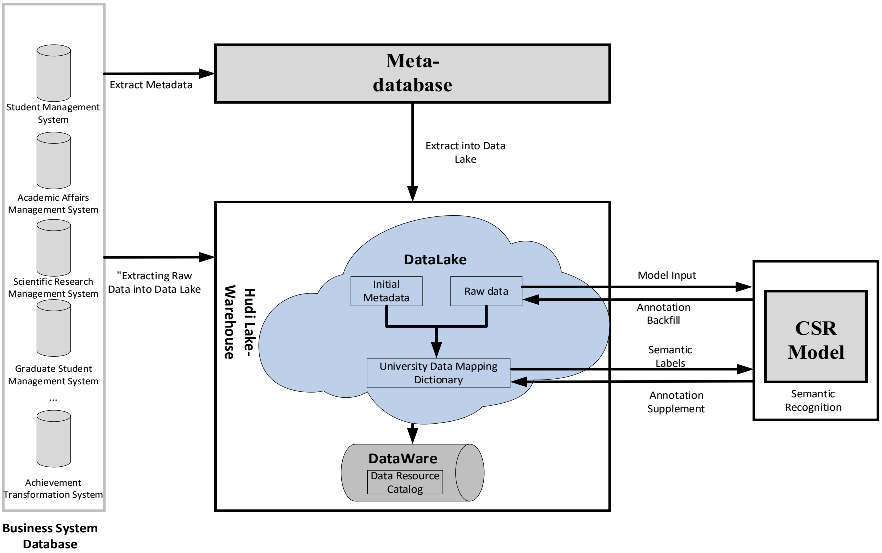
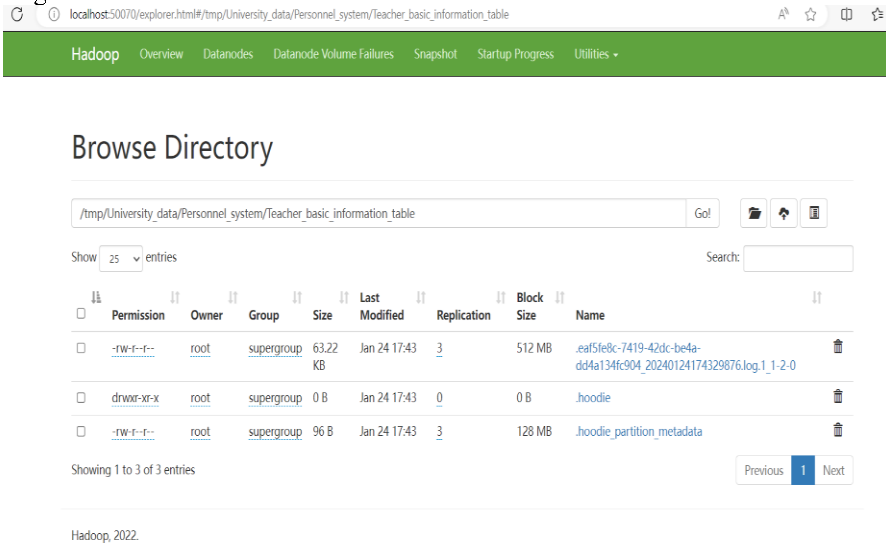


Fig. 1. Data Resource Directory Construction System Diagram

3.1 Data Lake Entry Research

The Hudi lake entry extraction utilizes the CDM (Cloud Data Migration) tool for bulk data migration, which achieves the structured data transfer from business system relational databases to the data lake through initial bulk extraction and batch incremental extraction methods. Additionally, CDL (Change Data Capture) real-time data access tool is used for capturing relational database change logs in real time and parsing them to generate commands for real-time operations like insertions, deletions, and modifications in the data lake records, thereby realizing real-time incremental migration of relational database data. The migrated data in the lake is transformed to Hudi's Merge On Read (MOR) format, using a mix of columnar file format (Parquet) and row-based file format (Avro) for data storage. Merge On Read stores immutable base data files in columnar Parquet format, while the incremental data files (Delta Files) resulting from new or modified data are stored in row-based Avro format. These incremental files are associated with base files and undergo a COMPACTION operation based on configurable strategies to merge real-time incremental data into columnar files.

For metadata extraction, the Datahub metadata management tool is chosen to connect to business system information and related extraction configurations. The database connection information includes the data source type, relevant data source configurations, and database extraction execution scheduling policies. A Python-written program is used for the DataHub's underlying database to standardize the initial metadata into dictionary format by splitting, normalizing, removing disordered characters, and then connecting with Hudi to implement Datahub metadata entry into the lake using bulk real-time methods. The diagram of data lake entry extraction is shown in Figure 2.



Browse Directory

/tmp/University_data/Personnel_system/Teacher_basic_information_table

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	-rw-r--r--	root	supergroup	63.22 KB	Jan 24 17:43	3	512 MB	.eaf5fe8c-7419-42dc-be4a-dd4a134fc904_20240124174329876.log.1_1-2-0
□	drwxr-xr-x	root	supergroup	0 B	Jan 24 17:43	0	0 B	.hoodie
□	-rw-r--r--	root	supergroup	96 B	Jan 24 17:43	3	128 MB	.hoodie_partition_metadata

Showing 1 to 3 of 3 entries

Hadoop, 2022.

Fig. 2. Data Lake Entry Extraction Diagram

3.2 Research on University Data Mapping Dictionary Design

Research on University Data Mapping Dictionary Design[5] is based on the GB/T 29808 national standard. It involves constructing a university data standard mapping dictionary and assimilating the Chinese abbreviations in its data standards into model semantic category labels. The standard names for data items, including data tables and data field items in the business system, are defined, serving as the corresponding standard names for metadata in the business system. The contents of the data mapping dictionary are shown in Table 1.

Table 1. The contents of the data mapping dictionary

Content	Explanation	Example
Data Standard Item Name	The Chinese standard abbreviation is concatenated in uppercase using the initial letters of the pinyin, and it serves as the data standard	XSXH
Chinese Standard Abbreviation	Data elements with semantic information from user-facing business systems, and use their Chinese abbreviations as semantic type labels in the model	Student ID Number
Standard Field Length	The maximum number of characters that a data item can contain	255
Standard Field Type	Data Types Contained in Data Items	VARCHAR

Standard Field Constraint	Description of Constraints on Data Items	PRIMARY KEY
Original Business System Name	Original Business System Name	Educational Administration System
Original Table Name	Table Name Corresponding to the Original Business System	Student Basic Information Table
Original Field Name	Field Name Corresponding to the Original Table	STU_ID
Original Field Length	Character Count Corresponding to the Original Field	255
Original Field Type	Data Type Corresponding to the Original Field	VARCHAR
Original Field Constraint	Constraint Status Corresponding to the Original Field	VARCHAR
Original Field Annotation A	Description Information Corresponding to the Original Field	PRIMARY KEY

Use the Chinese abbreviations from the data standard mapping dictionary as model semantic category labels, where semantic category labels are stored as strings in a list, with each semantic label being an independent string; associate the original field names with data standard item names in the university data standard mapping dictionary, achieving the mapping relationship between the databases of various university business systems and the university metadata standards.

3.3 Column Semantic Recognition

The text introduces the CSR (Column-Semantic-Recognition) model, which is a column semantic recognition model. It is based on the integration of lexical graph convolutional networks (GCN[6]) and RoBERTa[7], combining co-occurrence attribute interactions. The model captures global structural semantic information features and local structural semantic information features through a dual-layer GCN network and RoBERTa's multi-head self-attention mechanism, respectively. Additionally, it predicts classifications using an embedding AdaLine[8]adaptive strategy layer. The model also includes error correction mechanisms and incremental updates to optimize prediction results. The architecture of the CSR model is shown in Figure 3.

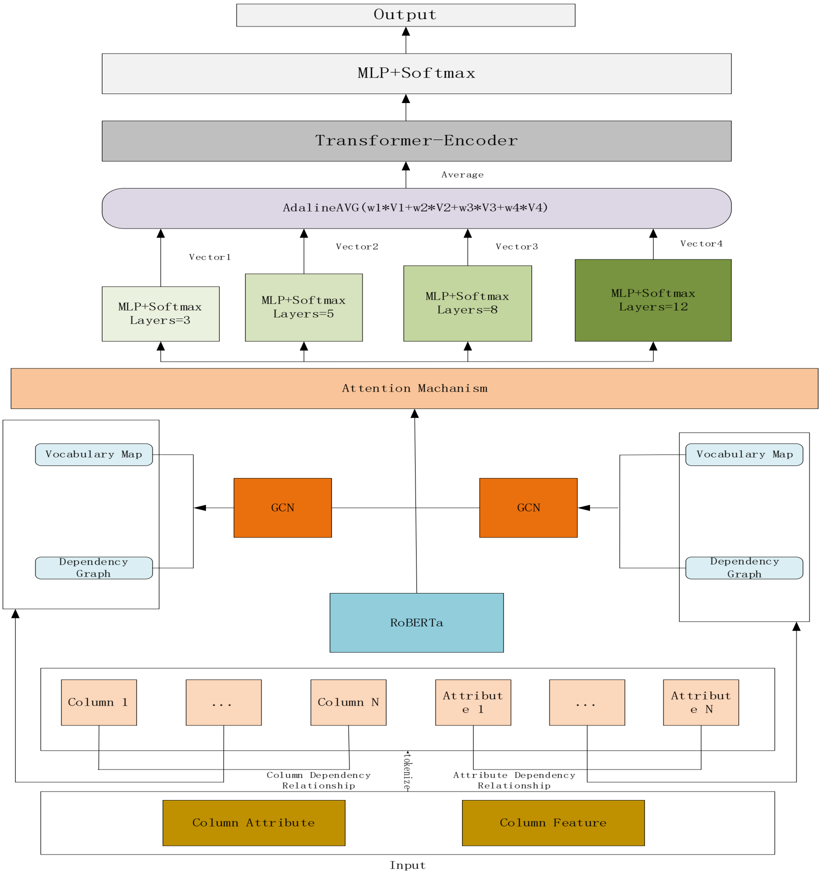


Fig. 3. CSR Model Architecture Diagram

GCN Global Semantic Feature Embedding.

The "GCN Global Semantic Feature Embedding Vector" refers to the feature vector that contains global contextual semantic information. This vector is the output after convolution through a dual-layer GCN (Graph Convolutional Network) graph convolution network.

(1)Constructing a Lexical Relationship Graph

1)Constructing a Lexical Graph

The text outlines a method for constructing a lexical graph using the WordNet semantic dictionary and NPMI [12](Normalized Pointwise Mutual Information). This process involves assessing the semantic relevance between word-word node pairs to build a large, heterogeneous lexical graph containing word nodes. Initially, the method calculates the weight between two word nodes using NPMI and determines the similarity of word nodes through WordNet clustering. If the NPMI between two word nodes exceeds a predetermined threshold, a semantically relevant edge is established between them. If not, the method employs the Wu-Palmer Similarity (WUP) approach from WordNet, which measures semantic similarity based on path structures. This involves calculating the distance of each word node to their Lowest Common Subsumer (LCS) and then normalizing this measurement to derive a similarity score. If the NPMI between two words does not exceed the threshold but the WUP does, a semantically relevant edge is still formed between them. The formulas for calculating NPMI between word nodes i and j, and for WUP, are provided in Equations I and II respectively. This approach combines multiple methods to evaluate semantic relationships and similarity, facilitating the construction of a comprehensive and nuanced lexical graph.

$$\mathrm{NPMI}(\mathrm{i},\mathrm{j}) = -\frac{1}{\log \mathrm{p}(\mathrm{i},\mathrm{j})} \log \frac{\mathrm{p}(\mathrm{i},\mathrm{j})}{\mathrm{p}(\mathrm{i})\mathrm{p}(\mathrm{j})} \tag{1}$$

In Equation I i and j represent word nodes, $\mathrm{p}(i,j) = \frac{\#w(i,j)}{\#w}$, $\mathrm{p}(i,j) = \frac{\#w(i,j)}{\#w}$, $\#\mathrm{w}$ represents the total number of sliding windows, $\#w(i)$ represents the number of all sliding windows containing the word node i, $\#\mathrm{w}(i,j)$ represents the number of all sliding windows containing both node i and node j;

$$\mathrm{WUP}(\mathrm{i},\mathrm{j}) = \frac{(2*\mathrm{depth}(\mathrm{LCS}(\mathrm{i},\mathrm{j})))}{\mathrm{depth}(\mathrm{i})+\mathrm{depth}(\mathrm{j})} \tag{2}$$

In Equation II, LCS(i, j) represents the Lowest Common Subsumer of word nodes i and j. depth(LCS(i, j)) denotes the depth of this Lowest Common Subsumer, which is the length of the path from the root node to the LCS. depth(i) indicates the depth of word node i in the WordNet hierarchy, and depth(j) indicates the depth of word node j in the WordNet hierarchy.

2)Constructing a Dependency Graph

The text describes the use of the TF-IDF [14](Term Frequency-Inverse Document Frequency) algorithm to create a dependency graph is based on the principle that if a specific word frequently appears in one document but rarely in others, it is considered to have a strong distinguishing ability for categorization and is very important for the expression of the document. The reason for using the TF-IDF formula is that it can effectively reflect the importance of words in a single document and distinguish them in a collection of multiple documents. Here, TF refers to the frequency of a specific word appearing in a given document. This number is usually normalized by dividing the count of the specific word by the total number of words in the document. IDF represents the importance of a specific word across the entire collection of documents. The IDF of a specific word can be calculated by dividing the total number of documents by the number of documents containing that word, and then taking the logarithm of this quotient. If the TF-IDF value exceeds a set threshold, a semantic dependency edge is created between the word and the document node; if the TF-IDF value is below the threshold, no edge is created. The word-document nodes are then weighted accordingly, resulting in the final dependency graph. The formula for calculating the TF-IDF value is provided in Equation III.

$$\mathrm{TF}-\mathrm{IDF} = \mathrm{TF} * \mathrm{IDF}(\mathrm{i}) = \frac{\mathrm{n}_{\mathrm{i},\mathrm{j}}}{\sum_{\mathrm{k}} \mathrm{n}_{\mathrm{k},\mathrm{j}}} * \log \frac{|\mathrm{D}|}{1+|\mathrm{j}:\mathrm{t}\in \mathrm{d}_{\mathrm{j}}|} \tag{3}$$

In Equation III, $\mathrm{n}_{\mathrm{i,j}}$ represents the frequency of occurrence of a specific word in the given document, $\sum_{k} n_{k,j}$ represents the total number of words in the document, $|D|$represents the total number of documents, and $|j : t_i \in \mathrm{d}_j|$represents the number of documents that contain the specific word.

3)Retrieving a Lexical Relationship Graph

Concatenate the similarity vectors from both the lexical graph and the dependency graph, and use a two-layer MLP (Multi-Layer Perceptron) for serial operation, transforming the vector into a form suitable for a dual-layer GCN (Graph Convolutional Network) graph convolution network. This results in a large, heterogeneous lexical relationship graph that captures the interactive semantic information of both the lexical and dependency graphs. The vector is then input into a dual-layer GCN for convolution operations, deriving the node's embedding vector based on the neighborhood properties of the node.

Firstly, for each node, a neighborhood computation graph is constructed, initializing the node vector representation at the 0th layer of the neighborhood computation graph as the node attributes. Subsequently, by aggregating the node information of the current layer and transferring the features to the next layer according to the hierarchical propagation rule, message passing from the 0th layer to the 2nd layer is conducted to obtain information from adjacent nodes. For a single convolution layer of the dual-layer GCN graph convolution network, sum the elements of the kth layer's nodes and then divide by the number of connections, effectively performing an element-wise averaging operation. The resulting vector is input into the dual-layer GCN graph convolution network for two layers of convolution operations. After passing through an activation function, the embedding of the k+1th layer v node is obtained, ultimately resulting in the node's GCN global semantic feature embedding vector.

$$\mathrm{H} = \mathrm{X}\tilde{\mathrm{A}}\mathrm{W} \tag{4}$$

$$\mathrm{GCN} = \mathrm{ReLU}(\mathrm{X}_{\mathrm{mn}}\tilde{\mathrm{A}}_{\mathrm{nn}}\mathrm{W}_{\mathrm{nh}}\text{）}\ \mathrm{W}_{\mathrm{hs}} \tag{5}$$

The formulas for the single-layer convolution followed by the dual-layer GCN graph convolution network are shown in Equations IV and V: In Equations IV and V, represents the lexical relationship graph of the dataset, W represents a hidden state of a weight of a single document, with dimensions |V|*h; m represents the batch size, n represents the size of the vocabulary, h represents the size of the hidden layer, and s represents the size of the sentence embedding.

Local Semantic Feature Embedding.

(1)Linearized Encoding

Dataset relation columns are concatenated row by row into a text segment, which is then tokenized and used as the input representation for the RoBERTa pre-trained model. After being encoded by the RoBERTa pre-trained model's Embedding, a preliminary column vector is output. Since the number of rows in each relation data table is not uniform, and the training requires the rows to be shuffled, a larger batch size of RoBERTa is used with a fixed maximum of 512 rows. According to the same relation theme, each relation table is split into multiple tables, and then the column dependencies of the relation table are concatenated row by row for linearization.

(2)Obtain Contextual Local Embeddings

The obtained preliminary column vector is input into the Transformer's three-layer multi-head column attention mechanism, which projects into different representational subspaces. In each subspace, each matrix uses Q, K, V matrices to focus on and compute the interrelationships between the current word and all other words in the sentence, continually adjusting the weight of each word. This process enhances the vector representation of the current word's local features. Finally, the output from the three-layer multi-head column attention mechanism is concatenated to form a deeper level representation that includes contextual local embeddings. The formula for the Transformer's single-head self-attention mechanism is shown as VI, and the formula for the multi-head attention mechanism is shown as VII.

$$\mathrm{Attention(Q,K,V)} = \mathrm{Softmax}(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d_K}}})\mathrm{V} \tag{6}$$

$$\mathrm{Multi(Q,K,V)} = \mathrm{Concat(head_1, head_2, ... head_n)\ where\ head_i} = \mathrm{Attention(Qw_i^Q, Kw_i^K, Vw_i^V)} \tag{7}$$

In formula (VI), Q represents the query matrix, K represents the key matrix, V represents the original features, and QK^T is the dot product operation, $\sqrt{d_k}$ is the dimension size of K. In formula (VII), head_i represents the head of the attention mechanism, and n is the number of heads in the attention mechanism.

Information Fusion for Predictive Classification.

Through the GCN global semantic feature embedding vector and the RoBERTa local semantic feature embedding vector, the dot-product attention weight function in the self-attention mechanism is used to calculate weights. During the training process of the RoBERTa pre-trained model, the output vector is passed through an MLP (Multi-Layer Perceptron) fully connected layer and normalized by the Softmax function, ultimately outputting the probability of each sample belonging to a semantic category for predictive classification. To enhance the robustness of the MLP layer, this paper adopts the Adaline algorithm to integrate the output probabilities of semantic categories. A multi-path MLP approach is used, combining results from different numbers of fully connected layers into a vector, which serves as the input for the Adaline algorithm. The Adaline algorithm scores different MLP outputs based on the output probabilities of semantic categories. Outputs whose probabilities are closer to the label's output value receive higher scores. Next, based on this scoring distribution, sampling is conducted on a normal distribution, and different weights are applied to each score based on the sampling values. The cumulative value thus obtained determines the weighted score of the current sample, leading to the final classification probability of the sample.

Label Correction and Incremental Update Optimization.

Utilize the Self-Attention mechanism of the Encoder part of the Transformer model to correct co-occurrence errors between labels, ensuring each input label obtains a corresponding output vector, which is further classified and mapped to the real category labels. Additionally, optimize the model parameters through a triplet cross-entropy loss function by calculating the cross-entropy pairwise and performing average pooling operations. Also, optimize the model by extracting incremental data from the Hudi data lake at fixed time intervals. This involves writing Shell-related code to enable routine operation of the Torch model. The routinely obtained model is trained with incremental data, where the T+1 incremental model replaces the T incremental model as the inference model, and the model is cyclically updated and optimized with extracted incremental data.

For each pair of categories i and j, the cross-entropy loss formula is shown as formula IX:

$$L_{i,j} = (y_i \log(\hat{y}_i) + y_j \log(\hat{y}_j)) \tag{8}$$

Generate $\frac{N(N-1)}{2}$ cross-entropy loss values, perform average pooling on these cross-entropy loss values, and obtain the final triplet entropy loss as shown in formula (X):

$$L_{ternary} = \frac{1}{\frac{N(N-1)}{2}} \sum_{i,j} L_{i,j} \tag{9}$$

Finally, the semantic recognition metadata will be updated and inserted back into the segmented original metadata annotations in the data warehouse. It will also be added to the blank fields of metadata annotation A in the constructed university metadata standard mapping dictionary.

3.4 Data Resource Catalog Construction Technology Research

Based on the university metadata standard mapping dictionary within the data warehouse, the standard definitions of tables and fields from the original business systems will be categorized and integrated according to the standard features of the subject domain. This will lead to the creation of a hierarchical and categorized university unified data resource catalog. It will enable the publication and search of the data resource catalog for universities.

Hierarchical: Based on the delegated authority of each university's secondary departments, the university data is divided into three security levels for sharing and openness, providing an effective basis for data security. The first level is data that is open to the entire university without the need for approval. The second level is data that can be opened to the entire university or specific individuals after anonymization or approval. The third level is data that is temporarily not open to the entire university.

Classification: The definition of subject domains is based on the revised university metadata standard mapping dictionary and the initial metadata stored in the data warehouse. It encompasses 16 business systems and 55 relational tables, leading to the design and division of subject domains and subdomains. The subject domain divisions include: organizational management domain, personnel management domain, teaching management domain, financial management domain, financial management domain asset management domain, and service management domain.

Next, for each subject domain, you can build the corresponding data model, which includes fact tables and dimension tables. For example, in the Personnel Management subject domain, you can design fact tables for students and teachers, as well as dimension tables for colleges and majors.

Subsequently, you can use Hudi's API or command-line tools to create data tables in the Hudi data warehouse to store data for each subject domain. Choose an appropriate storage format (e.g., COW or MOR) and configuration options for each table. Load the relevant data from the fact tables and dimension tables into their respective data tables.

Finally, you can wrap typical queries and cross-cutting statistical interfaces, optimize performance based on indexing, and build a web project using "Spring Boot+Bootstrap" to provide users with a visual resource catalog query service.

4 Implementation and evaluation of relevant experiments

4.1 Experimental Results and Analysis

Column Semantic Recognition Experimental Results.

We conducted experiments using the dataset provided by Qilu University of Technology, which includes data from various departments of the university. The dataset was divided into a training set, a test set, and a validation set in a 6:2:2 ratio. It covers data resources from 16 departmental systems and includes 55 basic tables, with a total of 54 relationships, an average of 12 columns per table, and 266 semantic labels. The training set contains 26,882 samples, the validation set 8,690, and the test set 8,690. To verify the results of our model, we used three benchmark methods for comparison: the CCA [13] model, which extracts a large number of statistical features; the SCA[14]model; and the CAI-Correction[15]model. The CCA model considers only the single-column information of the column to be identified, while the CAI-Correction model uses a self-attention mechanism to learn relational table-level context information containing only local information. The evaluation metrics used were accuracy (A), recall rate (R), and weighted average (F1) scores. The results of the CSR model are shown in Table 2.

Table 2. The experimental results of the CSR model

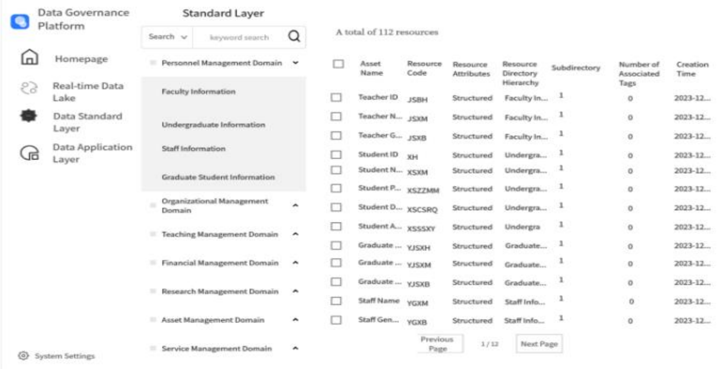
model	A	R	F1
CCA	92.2	72.55	91.16
Sherlock	95.8	87.66	95.82
CAI-Correction	97.3	86.73	94.87
CSR	**97.5**	**92.2**	**96.68**

From the experimental results, it can be seen that both the Co-occurrence Semantic Recognition (CSR) model and the CAI-Correction model perform well. Comparing the evaluation metrics with various baseline models, both CSR and CAI-Correction models, which consider context and self-attention mechanisms, demonstrate strong performance. However, the CSR model, which obtains global-local interaction context semantic features, outperforms the CAI-Correction model in terms of effectiveness.

Therefore, the CSR model we proposed shows the best performance, with an accuracy of 97.5%. It is comparable to the CAI-Correction model, which also considers contextual information from relationship tables, but outperforms the CCA model by 5.3% and the Sherlock model by 1.7%. Additionally, the CSR model achieves recall and weighted average scores of 92.2% and 96.68%, respectively, which are improvements over all three baseline methods. Compared to the CCA model, it has increased by 19.7% and 5.52%, and compared to the Sherlock model, it has increased by 4.6% and 0.86%. When compared to the CAI-Correction model, it has improved by 5.47% and 1.81%. In summary, the CSR column semantic model proposed in this paper is highly effective for relationship column semantic recognition tasks.

Data Catalog Construction Results.

This paper has constructed a unified university data resource catalog, which covers various domains including Organizational Management Domain, Personnel Management Domain, Teaching Management Domain, Financial Management Domain, Research Management Domain, Asset Management Domain, and Service Management Domain, as shown in Figure 4. Through a web-based visualization platform, the data resource catalog provides different data interfaces to various department heads responsible for different business departments. The querying options in the visualization platform can be categorized into simple queries and advanced queries. Simple queries require entering keywords related to the university data being queried, while complex queries involve setting multiple conditions for joint queries. Data users can precisely locate the data they need using these interfaces.



A total of 112 resources

□	Asset Name	Resource Code	Resource Attributes	Resource Directory Hierarchy	Subdirectory	Number of Associated Tags	Creation Time
□	Teacher ID	JSBH	Structured	Faculty In...	1	0	2023-12...
□	Teacher N...	JSXM	Structured	Faculty In...	1	0	2023-12...
□	Teacher G...	JSXB	Structured	Faculty In...	1	0	2023-12...
□	Student ID	XH	Structured	Undergra...	1	0	2023-12...
□	Student N...	XSXM	Structured	Undergra...	1	0	2023-12...
□	Student P...	XSZZMM	Structured	Undergra...	1	0	2023-12...
□	Student D...	XSCSRQ	Structured	Undergra...	1	0	2023-12...
□	Student A...	XSSSXY	Structured	Undergra	1	0	2023-12...
□	Graduate ...	YJSXH	Structured	Graduate...	1	0	2023-12...
□	Graduate ...	YJSXM	Structured	Graduate...	1	0	2023-12...
□	Graduate ...	YJSXB	Structured	Graduate...	1	0	2023-12...
□	Staff Name	YGXM	Structured	Staff Info...	1	0	2023-12...
□	Staff Gen...	YGXB	Structured	Staff Info...	1	0	2023-12...

Fig. 4. Data Catalog Construction Results

5 Conclusions

This paper presents a university relational data resource catalog construction technique based on a data lake and warehouse integration. It involves extracting raw data from various university business systems into the data lake using initial full-load, continuous batch increment, and real-time ingestion approaches. Historical data is loaded into the data lake to establish a comprehensive foundation of university data and its associated relationships. This approach ensures that data in the data lake remains synchronized with the data sources, ensuring real-time and accurate data. Furthermore, a university data standard mapping data dictionary is constructed, and the Chinese abbreviations in its data standards are assimilated into model semantic category labels. Additionally, metadata with labels is stored in the data lake relational table tags based on their relationships. The paper introduces the CSR model, which can merge global semantic information features and local structural semantic information features. It divides data generated without labels into different thematic data domains and builds data models for each subject domain. This results in the creation of a unified university data resource catalog, enabling data assets to be visualized and queried. This approach effectively overcomes the shortcomings of existing technologies, such as connectivity interference, the inability to promptly detect changes in the structure of raw system data, difficulty in interpreting metadata meanings, and lack of standardized data standards. It establishes a unified standard for the university data resource catalog, enabling the management, maintenance, analysis, and exploration of university data resources as data assets. It unlocks the value of data, promotes data sharing, and serves the purpose of university data governance.

Acknowledgments

This work is supported by the Key R&D Program of Shandong Province, China(No.2022CXGC020106),Pilot Project for Integrated Innovation of Science, Education, and Industry of Qilu University of Technology (Shandong Academy of Sciences(No.2022JBZ01-01).

References

1. Linlang L I U, Linzhen L U, Qinghong W U, et al. College Basic Development Status Data Management System Based on Data Governance Framework[J]. Journal of Donghua University (English Edition), 2023, 40(4).
2. Torabi F, Squires E, Orton C, et al. A common framework for health data governance standards[J]. Nature Medicine, 2024: 1-4.
3. Xue B, Zhu C, Wang X, et al. An Integration Model for Text Classification using Graph Convolutional Network and BERT[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 2137(1): 012052.

4. Errami S A, Hajji H, El Kadi K A, et al. Spatial big data architecture: From Data Warehouses and Data Lakes to the LakeHouse[J]. Journal of Parallel and Distributed Computing, 2023, 176: 70-79.
5. Ministry of education. Notice of the Ministry of Education on Issuing the Specifications for the Construction of Digital Campuses in Colleges and Universities (for Trial Implementation)[J]. 2023-09-13]. http://www. moe. gov. cn/srcsite A, 16.
6. H. Tang, Y. Mi, F. Xue and Y. Cao, "An Integration Model Based on Graph Convolutional Network for Text Classification," in IEEE Access, vol. 8, pp. 148865-148876, 2020, doi: 10.1109/ACCESS.2020.3015770.
7. Briskilal J, Subalalitha C N. An ensemble model for classifying idioms and literal texts using BERT and RoBERTa[J]. Information Processing & Management, 2022, 59(1): 102756.
8. Hou S, Wang N, Su B. Research on algorithm composition and emotion recognition based on adaptive networks[J]. Applied Mathematics and Nonlinear Sciences, 2023.
9. He J, Wang F, Ren H. The metadata management based on MongoDB for EAST experiment[J]. Fusion Engineering and Design, 2023, 195: 113955.
10. Sawadogo P, Darmont J. On data lake architectures and metadata management[J]. Journal of Intelligent Information Systems, 2021, 56: 97-120.
11. Kalyan K S, Rajasekharan A, Sangeetha S. Ammus: A survey of transformer-based pretrained models in natural language processing[J]. ar**v preprint ar**v:2108.05542, 2021.
12. Lee Y, Lee C, Lee H, et al. Normalizing Mutual Information for Robust Adaptive Training for Translation[C]//Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. 2022: 8008-8015.
13. Jalilifard A, Caridá V F, Mansano A F, et al. Semantic sensitive TF-IDF to determine word relevance in documents[M]//Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2. Singapore: Springer Singapore, 2021: 327-337.
14. Ding Y, Guo Y H, Lu W, et al. Context-aware semantic type identification for relational attributes[J]. Journal of Computer Science and Technology, 2023, 38(4): 927-946.
15. Gao S, YUAN W Z, Lu W, et al. Construction and Optimization of Co-occurrence-attribute-interaction Model for Column Semantic Recognition[J]. International Journal of Software & Informatics, 2023, 13(1).