Ding Y, Guo YH, Lu W *et al*. Context-aware semantic type identification for relational attributes. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 38(4): 927-946 July 2023. DOI: 10.1007/s11390-021-1048-y

Context-Aware Semantic Type Identification for Relational Attributes

Yue Ding[1, 2] (丁　玥), Yu-He Guo[1, 2] (郭雨荷), Wei Lu[1, 2, *] (卢　卫), *Member, CCF*
Hai-Xiang Li[3] (李海翔), *Member, CCF*, Mei-Hui Zhang[4] (张美慧), *Member, CCF, ACM, IEEE*
Hui Li[5] (李　晖), *Member, CCF, ACM, IEEE*, An-Qun Pan[6] (潘安群), *Member, CCF*, and
Xiao-Yong Du[1, 2] (杜小勇), *Fellow, CCF*

[1] *Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, Renmin University of China Beijing 100872, China*

[2] *School of Information, Renmin University of China, Beijing 100872, China*

[3] *Tencent (Beijing) Technology Company Limited, Beijing 100080, China*

[4] *School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*

[5] *College of Computer Science and Technology, Guizhou University, Guiyang 550025, China*

[6] *Tencent (Shenzhen) Technology Company Limited, Shenzhen 518057, China*

E-mail: dingy_96@ruc.edu.cn; guoyuhe@ruc.edu.cn; lu-wei@ruc.edu.cn; blueseali@tencent.com; meihui_zhang@bit.edu.cn cse.HuiLi@gzu.edu.cn; aaronpan@tencent.com; duyong@ruc.edu.cn

Received October 5, 2020; accepted June 9, 2021.

Abstract Identifying semantic types for attributes in relations, known as attribute semantic type (AST) identification, plays an important role in many data analysis tasks, such as data cleaning, schema matching, and keyword search in databases. However, due to a lack of unified naming standards across prevalent information systems (a.k.a. information islands), AST identification still remains as an open problem. To tackle this problem, we propose a context-aware method to figure out the ASTs for relations in this paper. We transform the AST identification into a multi-class classification problem and propose a schema context aware (SCA) model to learn the representation from a collection of relations associated with attribute values and schema context. Based on the learned representation, we predict the AST for a given attribute from an underlying relation, wherein the predicted AST is mapped to one of the labeled ASTs. To improve the performance for AST identification, especially for the case that the predicted semantic types of attributes are not included in the labeled ASTs, we then introduce knowledge base embeddings (a.k.a. KBVec) to enhance the above representation and construct a schema context aware model with knowledge base enhanced (SCA-KB) to get a stable and robust model. Extensive experiments based on real datasets demonstrate that our context-aware method outperforms the state-of-the-art approaches by a large margin, up to 6.14% and 25.17% in terms of macro average F_1 score, and up to 0.28% and 9.56% in terms of weighted F_1 score over high-quality and low-quality datasets respectively.

Keywords attribute semantic type (AST) identification, context-aware, semantic embedding, knowledge base embedding

1 Introduction

Given a relation R, attribute semantic type (AST) identification targets to identify the semantic types of attributes in R. AST identification has a wide range of applications in data cleaning[1], schema matching[2, 3], keyword search in databases[4] and so on. For example, in schema matching that aligns attributes with the same meanings from multiple relations, AST identification is used to first figure out the meaning of

Regular Paper

The work was supported by the National Key Research and Development Program of China under Grant No. 2020YFB2104100, the National Natural Science Foundation of China under Grant Nos. 61972403 and U1711261, the Fundamental Research Funds for the Central Universities of China, the Research Funds of Renmin University of China, and Tencent Rhino-Bird Joint Research Program.

*Corresponding Author

each attribute, denoted as the semantic type, in relations.

Processing AST identification poses some challenges. First, due to a lack of unified naming standards across prevalent information systems, attributes of relations in the databases are often named at will. Therefore, it is challenging to figure out the semantic types of attributes since attributes with different names can be referred to as the same semantic types, while attributes with the same names can be referred to as different semantic types. Second, in many cases, like web tables, attributes in the relations do not have explicit names. In these cases, it is not applicable to figure out the semantic types of attributes based on the attribute names. Third, the same attribute under different kinds of context can be even referred to as quite different semantic types due to the finer granularity. For example, *name* under the teaching context refers to teachers' names, while *name* under the studying context refers to students' names. However, in such cases, it is challenging to infer the semantic types only based on the name attributes.

Thus far, existing approaches that process AST identification are divided into three categories, rule-based approaches, knowledge-based approaches, and feature-based approaches. Rule-based approaches[5–7] solve the problem by defining specific rules over the attribute values and doing regular expression matching or dictionary lookup to identify ASTs. Rule-based approaches are applicable to well-formed attribute values, e.g., *e-mail address* and *gender*, and they suffer from regular attribute values, which are often the cases in reality. Alternatively, knowledge-based approaches[4, 8–10] utilize external knowledge from the Internet or existing knowledge bases, and develop reasoning algorithms to facilitate the judgment of ASTs. It is worth mentioning that, due to the gap between the applications and knowledge bases, it is often the case that a corresponding match cannot be found in knowledge bases to assist AST identification, especially for the labeled attribute types with few instances[10]. More flexibly, feature-based approaches[11–13] abstract the syntax and semantic characteristics based on the attribute values and apply the classification models to identify ASTs with similar or various characteristics. Although a variety of feature-based approaches have been proposed to improve the effectiveness of AST identification, there are still two major challenges to be tackled. On one hand, as discussed above, the same attribute with different kinds of context can be referred to as quite different semantic types, like students' name and teachers' names. It is difficult to distinguish them using existing attribute-wise feature-based approaches. On the other hand, most feature-based approaches transform the AST identification problem to a multi-class classification problem; however, they suffer from the unknown semantic types, i.e., the semantic type of the predicted attribute does not exist in the labeled attribute semantic types.

In this paper, we study the above issues and propose a context-aware method for the AST identification based on schema context aware semantic embeddings and knowledge base embeddings. The main contributions of our paper are as follows.

• Considering that attributes in the same relation are often mutually helpful to AST identification, we propose a schema context aware (SCA) model, which generates semantic embeddings from a collection of relations associated with attribute values and schema context and performs well on assigning the semantic type for a given attribute.

• To enhance the representation and alleviate the problem that the predicted semantic types of attributes may not be labeled, we extract embeddings with reference to entities and ontology classes in knowledge bases and construct a schema context aware model with knowledge base enhanced (SCA-KB) model. For attributes with unknown types, the knowledge base embeddings are used for candidate types generation.

• Extensive experiments over several real datasets demonstrate that our SCA-KB model outperforms the recently state-of-the-art approaches by a significant margin. We can choose the corresponding ensemble approach in the SCA-KB model according to different data characteristics.

The remainder of this paper is structured as follows. Section 2 provides a literature review of AST identification. Section 3 formalizes the AST identification problem to be studied in this paper. Section 4 introduces the architecture of our proposed context-aware model in detail. Section 5 includes the experimental datasets, model implementation details and evaluation metrics. Experimental results and findings are presented in Section 6 to demonstrate the excellent performance of our approach. Finally, Section 7 summarizes this paper and gives the future work directions.

2 Related Work

AST identification for relational attributes has been studied for several years. Broadly speaking, existing approaches are divided into three categories: rule-based approaches, knowledge-based approaches, and feature-based approaches.

2.1 Rule-Based Approaches

Rule-based approaches answer the AST identification problem by defining domain-specific rules over the attribute values and doing regular expression matching or dictionary lookup to identify ASTs. These approaches are applicable to well-formed attribute values, e.g., credit code, e-mail address, gender and zip code.

Some existing commercial data preparation and analysis systems (e.g., Google Data Studio[5], Microsoft Power BI[6], and Trifacta[7]) define heuristic rules for AST identification. Based on the expert experience for type customization, the rule-based approaches can identify attribute types efficiently on limited attributes. Considering the variety of ASTs and data expressions in relations, the rule-based approaches lack generality and scalability.

2.2 Knowledge-Based Approaches

Knowledge-based approaches process AST identification by mapping the attributes of the relations to the entity types extracted either from Internet on the fly or from knowledge bases (KBs)[14].

Venetis *et al.*[4] built a mapping from attributes to a pre-defined set of labels based on the knowledge extracted from Internet. With Bing's knowledge graph[15], Zhao and He[8] built the mapping based on the collected entity types and rich synonymous names of known entities. Furthermore, using entities and ontology classes stored in existing KBs, e.g., DBPedia[16], Freebase[17], YAGO[18], AST identification could be resolved by referring to the correspondences between attributes and ontology classes with strategies like majority voting[19]. Chen *et al.*[9] developed a novel KB lookup and reasoning algorithm for property feature extraction which indicates potential relations between attributes and provided discriminative predictive information.

In reality, the above approaches make an implicit assumption that each attribute value in relations has a one-to-one mapping via exact matching to an entity in the knowledge base. However, due to the so-called knowledge gap, introduced by typos or a lack of unified naming standards, it is often the case that we cannot find a proper entity simply based on the underlying attribute values. Differing from knowledge-based approaches, in this paper, we utilize entities in the knowledge base, i.e., DBpedia, to generate partial attribute features to predict the AST types instead of the exact match to address this issue.

2.3 Feature-Based Approaches

Attribute values with the same semantics tend to have similar syntax and semantic characteristics. For this reason, feature-based approaches typically first extract semantic characteristics of the attribute and then apply classification models to process the AST identification problems based on the semantic characteristics.

Ramnandan *et al.*[11] used Kolmogorov-Smirnov (K-S) test and Term Frequency-Inverse Document Frequency (TF-IDF) to characterize the numerical and textual data respectively. Furthermore, Pham *et al.*[12] introduced the Mann-Whitney test and Jaccard similarity to characterize numerical data and textual data, and then trained the logistic regression and a random forest model for AST identification. As a major branch of machine learning, deep learning (DL) yields state-of-the-art achievements for predictive tasks in multiple domains[20–23]. Sherlock[13] extracts high-dimensional features including character distributions, pre-trained word embeddings, self-trained paragraph embeddings and column statistics from each attribute, and trains a multi-input deep neural network for type prediction. Chen *et al.*[9] embedded the phrase within a cell with a bidirectional recurrent neural network and an attention layer (Att-BiRNN) and learned attribute features and row features with a convolutional neural network (CNN).

Feature-based approaches answer the AST identification problem in an attribute-wise paradigm. Nevertheless, in reality, attributes in the same relation are often mutually helpful for AST identification. For this reason, we propose to answer the AST identification problem in a relation-wise paradigm, i.e., we generate relation-wise embeddings with the consideration of schema context to predict AST in relations.

3 Problem Statement

We use symbol R to denote an underlying relation which consists of n attributes, where each attribute is denoted as a_i, $1 \leqslant i \leqslant n$. Besides, let symbol $\boldsymbol{A}_i$ represent the list of values for attribute a_i in relation R and $\boldsymbol{A}^{\rm S}$ represent the attribute value matrix of relation R, i.e., $\boldsymbol{A}^{\rm S} = (\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n)$. For our purpose, we denote the semantic type of attribute a_i as t_i, where t_i is selected from a predefined attribute semantic type collection T.

Problem Statement. Given a relation R with attribute value matrix $\boldsymbol{A}^{\rm S}$, the problem of AST identification is to figure out t_i for each a_i in R.

4 Context-Aware Method for Semantic Type Identification

According to the problem statement in Section 3, we can conclude that when the set T of ASTs is predefined, the AST problem can be transformed into a multi-class classification problem. In this section, we introduce our context-aware method for semantic type identification. Firstly, data preprocessing rules are defined in Subsection 4.1 to generate candidate semantic type set and high quality attribute-semantic-type-labeled relations. In Subsection 4.2, we propose a schema context aware (SCA) model to learn embeddings from a collection of relations associated with attribute values and schema context and figure out ASTs for attributes based on the embeddings. Further, to improve the performance for AST identification, especially when the predicted semantic types of attributes are not included in the candidate semantic type set, we map attribute values to corresponding entities in the knowledge base for KB embedding generation (Subsection 4.3) and construct a schema context aware model with knowledge base enhanced (SCA-KB) to get a stable and robust model (Subsection 4.4).

4.1 Data Preparation

We formulate the AST identification problem as a classification task. The unlabeled target attribute in relations will be mapped to the most likely matching semantics in the predefined candidate type set T. Therefore, the definition of types is particularly significant for the specific dataset. In this paper, we pre-define the disjoint candidate types and ground truth labels of attributes by referring to the metadata like attribute names of corresponding relational datasets.

In addition, in order to adapt to case differences in different datasets flexibly, we uniformly identify attribute types as lowercase, for example, converting "*Date*", "*DATE*" and other similar writing formats to "*date*" case-insensitively. For attribute names consisting of multiple words, we remove the spaces between the words and concatenate them with the connecting line, such as "*birth Date*" ⇒ "*birth_date*".

According to the type definition criteria mentioned above, the standardized candidate semantic type set T can be determined manually. Labeling attributes with reference to raw attribute names in relations and the corresponding standard type in T yields high-quality attribute-semantic-type-labeled relations. The processed data examples are shown in Table 1, Table 2 and Table 3.

4.2 Semantic Embedding with Pre-Trained Model

The state-of-the-art approach, Sherlock[13], extracts attribute-wise features for AST identification but cannot work well on identifying attributes with similar characteristics but different ASTs. Considering that attributes in the same relations are often mutually helpful to AST identification, we introduce a pre-trained language model BERT (Bidirectional Encoder Representations from Transformers) to generate semantic embeddings based on both attribute values and schema context.

4.2.1 Column Content Aware Model

BERT borrows its structure from transformer[24]. The architecture of bare BERT is an encoder with 12 identical stacked layers, with each layer taking outputs of the former layer as inputs. Each layer is a transformer block, making up of two sub-layers, including a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Each sub-layer is wrapped by residual connection[25]

Table 1. Example Book Relation for Attribute Semantic Type Identification

title	*author*	*date*	*price*	*publisher*
Echoes from Lane Field	Bill Swank	1-Jun-99	\$16.99	Turner Publishing Company
A's Essential	Steve Travers	1-Apr-07	\$9.99	Triumph Books
Endless Summers	Jack Torry	1-Mar-96	\$14.99	Taylor Trade Publishing

Table 2. Example Production Relation for Attribute Semantic Type Identification

name	*brand*	*price*
MacBook Pro 15.4	Apple	\$3 599.00
ThinkPad Helix 37014DU	Lenovo	\$2 975.22
Alienware 17 ANW17 17.3	Dell	\$2 599.00

and followed by layer normalization[26]. The output of each sub-layer can be represented as (1).

$$LayerNorm(\boldsymbol{X} + Sublayer(\boldsymbol{X})), \tag{1}$$

where $Sublayer(\boldsymbol{X})$ is the function implemented by the sub-layer itself and $\boldsymbol{X}$ is the input matrix of this sub-layer.

Overall, the self-attention mechanism is the most important part in BERT. Different from RNN, which is often employed in sequence modeling, the self-attention mechanism avoids computation along the input and output sequences, with one hidden state summarizing precedent inputs in evolution step by step. That is, when calculating an output token vector $\boldsymbol{z}_i$, self-attention allows all vectors outputted from the former layer to be attended and summed up with weight coefficients $\boldsymbol{\alpha}_i$ altogether.

$$e_{ij} = \frac{(\boldsymbol{E}_i \cdot \boldsymbol{W}^{\mathrm{Q}}) \cdot (\boldsymbol{E}_j \cdot \boldsymbol{W}^{\mathrm{K}})^{\mathrm{T}}}{\sqrt{d_z}}, \tag{2}$$

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{n} \exp e_{ik}}, \tag{3}$$

$$\boldsymbol{z}_i = \sum_{j=1}^{n} \alpha_{ij} (\boldsymbol{E}_j \cdot \boldsymbol{W}^{\mathrm{V}}). \tag{4}$$

In (4), $\boldsymbol{W}^{\mathrm{V}}$ is the parameter matrix transforming $\boldsymbol{E}_i$ and $\boldsymbol{E}_j$. Note that $\boldsymbol{E}$ is the input matrix of the current attention head. Each weight coefficient α_{ij} is computed using a softmax function (3), where e_{ij} is computed by comparing the similarity of two input elements by dot production (2) and d_z is the dimension of the input and output vector in attention heads. The division operation is known as scaled dot product. $\boldsymbol{W}^{\mathrm{V}}$, $\boldsymbol{W}^{\mathrm{Q}}$ and $\boldsymbol{W}^{\mathrm{K}}$ are all parameter matrices for linear transformation. The three matrices are unique for each attention head.

As mentioned in Section 3, with the definition of ASTs and the acquisition of high-quality attribute-semantic-type-labeled training data, we transform the problem of AST identification into a classification problem. To map an attribute to the vector space for semantic embedding and achieve good performances on AST identification, we focus on fine-tuning based on the pre-trained BERT model for AST classification which is the downstream task we defined. We add a projection layer wrapped by tanh activation and a classification layer on the top of BERT to compute the label probabilities (5).

$$\boldsymbol{P} = \mathrm{softmax}(\tanh(\boldsymbol{C} \cdot \boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2), \tag{5}$$

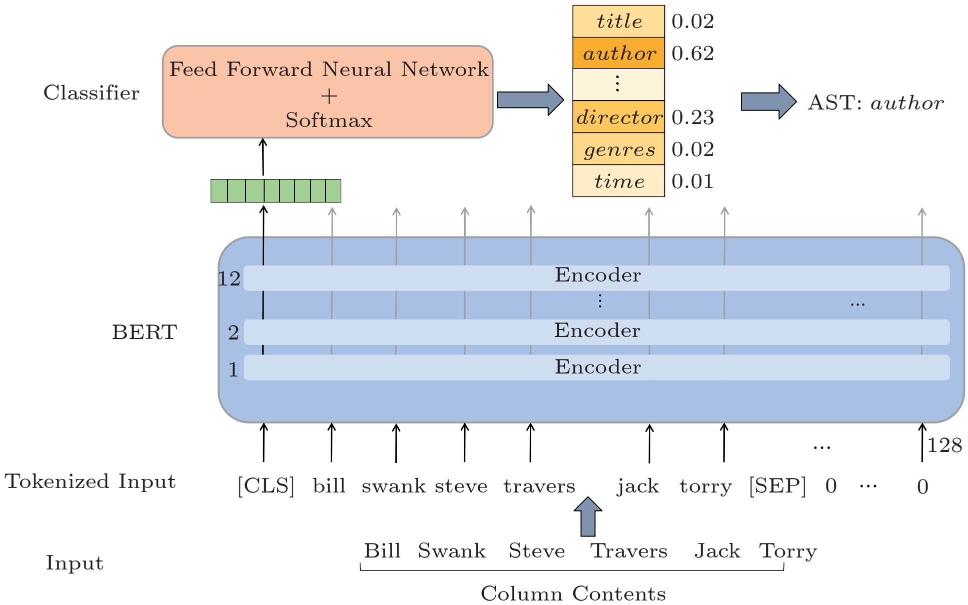
where $\boldsymbol{C}$ is the output [CLS] representation, $\boldsymbol{W}_1 \in \mathbb{R}^{d_z \times d_z}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{d_z \times n_{\mathrm{labels}}}$ are matrices for linear transformation, and $\boldsymbol{b}_1 \in \mathbb{R}^{d_z}$ and $\boldsymbol{b}_2 \in \mathbb{R}^{n_{\mathrm{labels}}}$ are corresponding biases. All the parameters of BERT are fine-tuned based on our training set with cross entropy.

As the input token sequence to BERT may be a single sentence or sentence pair packed together and the "sentence" here can be an arbitrary span of contiguous text, rather than an actual linguistic sentence[27], we concatenate the attribute values to form a sequence inputted to BERT and add a special classifier token [CLS][28, 29] at the start of input value sequence. All the tokens including [CLS] go through multiple self-attention layers in a fine-tuned BERT to get higher-level representation regarding to all other co-occurring tokens. Besides, for each layer, [CLS] attends to all the positions of the input sequence and sums them together, which can be regarded as a summarization of representations from the former layer. Finally, at the top layer, [CLS] is chosen for semantic embedding of the inputted attribute. With the raw representation of the training dataset, the fine-tuning of the predefined neural network is performed, so as to adjust the parameters of the model. The architecture of the column content aware (CCA) model is shown in Fig.1(a)①. Briefly speaking, the input and output of the model are shown in Fig.1(b).

Table 3. Example Movie Relation for Attribute Semantic Type Identification

movie	*year*	*director*	*genres*	*time*
High-Rise	2015	Ben Wheatley	Action, Drama, Sci-Fi	112 (min)
Mercy for Angels	2015	K.C. Amos	Action, Drama, Thriller	91 (min)
Barely Lethal	2015	Kyle Newman	Action, Adventure, Comedy	96 (min)

①In BERT, wordpieces are actually used as tokens instead of words. For ease of understanding, the figure is simplified for labeling.



(a)

Input:

Column Contents: {Bill Swank, Steve Travers, Jack Torry}

Output:

AST: *author*

(b)

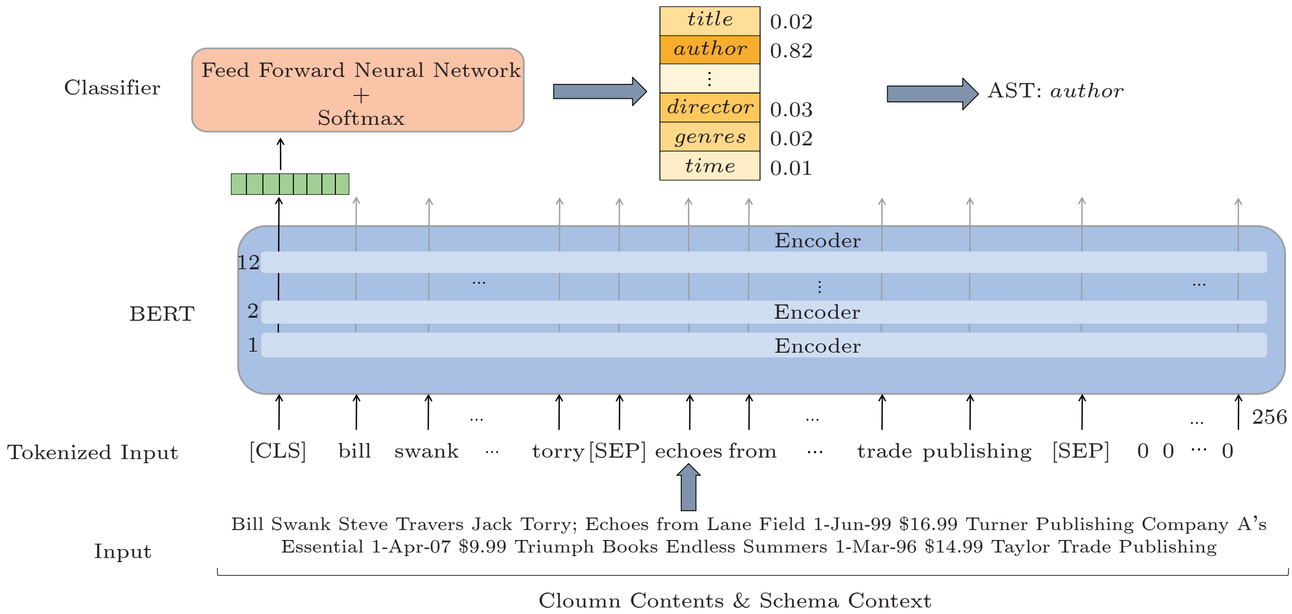
Fig.1. Column content aware model. (a) Model architecture. (b) Input and output example.

4.2.2 Schema Context Aware Model

Simply utilizing the attribute values is ambiguous to identify attribute semantics. Since attributes with similar values can be referred to as quite different semantic types, e.g., the *author* in Book (Table 1) and the *director* in Movie (Table 3). The actual semantic type depends on the theme of the whole relation, which is inherent when the schema is defined and can be revealed from the values of its co-occurring attributes. To tackle this problem, we enhance the CCA model by leveraging the schema information, namely schema context aware (SCA) model, to get more representative embeddings and identify ASTs precisely.

The SCA model is motivated by the fact that the context plays an important role in word representation learning. For all the word embedding models, their training objectives implicitly follow the Harris' Distributional Hypothesis[30] that can be stated as follows: words that occur in similar contexts tend to have similar meanings[31], wherein "context" is regarded as the co-occurring words which precede and follow the target word within some distance. We extend the context in word representation learning to relations and introduce the schema context. For an attribute with its semantics to be predicted, there often exist many other attributes co-occurring in a single relation, and we then extend the Harris' Distributional Hypothesis below: "Attributes that occur in similar schema contexts tend to have similar semantic types". The schema context for an attribute we defined refers to attributes co-occurring in the same relation. This simplification is useful especially when correlations between attributes are unknown in web tables.

The architecture of the SCA model is shown in Fig.2(a). As mentioned above, the input token sequence to BERT can be two sentences packed together. To combine the information of schema context, we just need to keep the structure of the CCA model and add one more "sentence" formed by concatenating the schema context to the input. Two parts of information, including the single column content and its schema context separated by a special token [SEP], are fed into the cross encoder BERT, and a target value is predicted. Briefly speaking, the input and



(a)

Input:

Column Contents: {Bill Swank, Steve Travers, Jack Torry}
Schema Context:
{Echoes from Lane Field, 1-Jun-99 $16.99, Turner Publishing Company},
{A's Essential, 1-Apr-07 $9.99, Triumph Books},
{Endless Summers, 1-Mar-96 $14.99, Taylor Trade Publishing}.

Output:

AST: *author*

(b)

Fig.2. Schema context aware model. (a) Model architecture. (b) Input and output example.

output of the model is shown in Fig.2(b).

After fine-tuning with the downstream task of classification, a well-performed classifier can be obtained and representations generated during classification are high-quality semantic embeddings for attributes. With comparison, the advantages of our SCA model are as follows.

1) Adopting a fine-tuned BERT as feature extractor makes the entire training process become an end-to-end training mode. Furthermore, with WordPiece tokenization (Byte Pair Encoding, BPE) adopted, the problem of out-of-vocabulary (OOV) words is solved, which is severe in Sherlock's[13] feature extraction procedure, since missing values may yield when using GloVe[32] for word embeddings or using PV-DBOW[33] for paragraph embeddings.

2) With the collection of attribute values and schema context, our model generates relation-wise representations which implicit the theme of relations. By doing this, attributes with similar values but quite different semantics can be identified more precisely.

4.3 Knowledge Base Embedding with Entity Retrieval

In addition to characterizing the semantics of attributes with the language model, we create knowledge base embedding (a.k.a. Knowledge Base Vector, KBVec) with reference to knowledge base containing complex structured and unstructured information. KBVec utilizes prior knowledge in the KB to express potential type characteristics of attribute values. Each slot of KBVec indicates the possibility of the sample attribute belongs to a specific KB ontology class, and the dimension of the raw KBVec depends on the number of ontology classes. The classes defined by KB have hierarchical relationships, e.g., *actor*, *athlette*, *chef*, and *dancer* are sub-classes of *person*. Therefore, in most cases, the values of each attribute will be mapped to several ontology classes with parent-child relationships, and the generated KBVec will be high-dimensional and sparse. We further use Principal Component Analysis (PCA) to reduce the dimensionality of the feature space and minimize the feature loss during dimensionality reduction.

The extraction algorithm of KBVec is shown in Algorithm 1. For each attribute value in the input attribute, the algorithm first retrieves the matched entities in the knowledge base, and obtains all ontology classes (including all parent and children classes) it belongs to (line 5). As the look-up by lexical matching is ambiguous, the maximum number of returned results is set to N to avoid missing the right entity. After obtaining the candidate classes that the attribute value belongs to, we will retrieve the slot ID of each candidate class in KBVec, and enhance the representation in the corresponding slot (lines 6–9). Finally, we apply normalization to facilitate further processing and obtain the raw KBVec characterizing the category features of attributes. In order to reduce the noise interference caused by high-dimensional embeddings, PCA is used to perform dimensionality reduction (lines 13 and 14). Note that we adopt the same PCA model for the training, validation and test set in the same dataset, i.e., transforming the embeddings to the consistent dimension for further construction and application of the classification model.

Algorithm 1. Extraction of KBVec

Input: attributes with values $Attrs$, ontology classes $clses$ with the size of d, a maximum number of match entities N

Output: $\boldsymbol{KBVec}$

```
1  Vector = [ ]
2  for every attr in Attrs do
3      Initialize the d-dimensional V for attr with zero
4      for every val in attr do
5          Look up N CandidateClasses for val in KB
6          for every cls in CandidateClasses do
7              Find the index of cls in clses
8              Add 1 in the corresponding slot of V
9          end for
10         Append V to Vector
11     end for
12 end for
13 Apply z-score normalization to Vector
14 Perform dimensionality reduction with PCA to get
   KBVec
15 return KBVec
```

4.4 Schema Context Aware Model with Knowledge Base Enhanced

In summary, we obtain the high-quality semantic embedding with the SCA model, and extract the representative knowledge base embedding with the designed KBVec extraction algorithm.

To improve the identification performance for both the predefined types and the unknown types, we conduct ensemble modeling and construct the SCA-KB model. There are three ensemble approaches to be considered.

Score Ensemble. Based on the average word vector ($\boldsymbol{AvgWV}$) of the attribute values and $\boldsymbol{KBVec}$, a basic multi-class classifier with logistic regression (LR) can be constructed[②]. Applying both the SCA model and the basic classifier for AST identification, we get two prediction results $\boldsymbol{P}$ and $\boldsymbol{P}_{\rm basic}$. The score ensemble approach takes an average of them for the final decision $\boldsymbol{P}_{\rm result}$ (6).

$$\begin{aligned}&\boldsymbol{P}_{\rm basic}=\mathrm{LR}([\boldsymbol{AvgWV},\boldsymbol{KBVec}]),\\&\boldsymbol{P}_{\rm result}=(\boldsymbol{P}+\boldsymbol{P}_{\rm basic})/2.\end{aligned}\tag{6}$$

Feature Ensemble with LR. With the conventional feature fusion approach, we concatenate the extracted fine-tuned semantic embedding $\boldsymbol{C}$ (i.e., the output [CLS] representation) with knowledge base embedding $\boldsymbol{KBVec}$, and train a multi-class classifier with LR to get the prediction probabilities (7).

$$\boldsymbol{P}_{\rm result}=\mathrm{LR}([\boldsymbol{C},\boldsymbol{KBVec}]).\tag{7}$$

Feature Ensemble with BERT. In this ensemble approach, $\boldsymbol{KBVec}$ is concatenated with the generated semantic embedding in the SCA model during fine-tuning, so as to adjust the parameter of the model. Compared with the model architecture mentioned in Subsection 4.2, one more projection layer wrapped by tanh activation is added to convert the fusion vector to the original dimension. The prediction probability can be represented as (8).

$$\begin{aligned}\boldsymbol{P}_{\rm result}=\ &\mathrm{softmax}(\tanh(\tanh([\boldsymbol{C},\boldsymbol{KBVec}]\boldsymbol{W}_0+\\&\boldsymbol{b}_0)\boldsymbol{W}_1+\boldsymbol{b}_1)\boldsymbol{W}_2+\boldsymbol{b}_2),\end{aligned}\tag{8}$$

where $\boldsymbol{W}_0$, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are all matrices for linear transformation, and $\boldsymbol{b}_0$, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are their corresponding biases. All of the parameters are fine-tuned based on our training set.

5 Experimental Setup

We conduct extensive experiments to verify the effectiveness of our method.

[②]LR is a binary classification model which does not support the multi-class classification natively. Thus we adopt the one-versus-one strategy to split a multi-class classification into one binary classification problem for each pair of classes.

5.1 Datasets

Three datasets with different predefined semantic types are chosen to verify the performance of our model and the ground truth labels of attributes are generated with reference to the metadata like attribute names. Multiple samplings based on the datasets yield our experimental datasets which are split into the training, validation and test set with the ratio of $3:1:1$. More details can be seen in Table 4.

Benchmark Dataset. The benchmark dataset (BMdata)[34] for entity resolution describing information about bibliography and e-commerce has canonical and common schemas which can be used as the ground truth labels of ASTs. According to the data preparation process mentioned in Subsection 4.1, eight common types including *author*, *description*, *manufacturer*, *name*, *price*, *title*, *venue*, and *year* are defined.

Magellan Dataset. With the expansion of BMdata, the Magellan dataset contains more relations extracted and converted from data-rich websites[35]. The dataset includes relations from multiple application domains, e.g., anime, beer, bibliography, bike, book, e-commerce, movies, music, and restaurants. Similarly, the Magellan dataset frequently used for entity resolution is of good quality, and thus the semantic type set can be defined referring to the attribute names in original relations easily. As a result, 71 common semantic types are defined after filtering.

Web Table Corpus. Except for the high-quality relational data published by researchers, there are vast amount of relations embedded in the web providing extensive information on entities from various domains. Referring to the web tables (WebTable) extracted from a couple thousand of websites[36], we obtain 38 493 relations and define 83 informative semantic types. Considering the multi-domain and low-quality features of web tables, we explore whether our model can perform well on identifying ASTs which occur in diverse schema contexts.

The above datasets include both well-structured data obtained from existing research work and relatively low-quality data captured from Internet full-text documents, which have heterogeneous representations, misspellings and extraction errors. We evaluate the performance of our AST identification approach based on the three datasets.

Generally speaking, the number of samples in each semantic type varies a lot. As shown in Fig.3 which takes the Magellan dataset as an example, the imbalances of per-type sample numbers form the "long-tail" distributions, where the common attribute types occur frequently in different contexts while the rare types are only contained in specific context. The AST identification model should capture and distinguish the differences between various types regardless of the imbalance support for respective types.

5.2 Model Implementation

For semantic embedding, we implement the CCA model and the SCA model based on the pre-trained uncased BERT base model with PyTorch as backend, and tune the hyper-parameters on each dataset respectively to construct the specific target model. During fine-tuning, most of the hyper-parameter settings are kept the same as pre-trained in BERT, except for the batch size, learning rate, and the number of epochs. Devlin *et al.*[27] pointed out that almost all tasks work well with a small batch size, a small learning rate, and a few training epochs. For optimal performance, we fine-tune the BERT model on three TITAN RTX GPUs with the batch size of 32, learning rate of 2.0×10^{-5}, and save the best model on the validation set during three training epochs. In addition to the setting of above parameters, the maximum sequence length is set variously depending on the specific task. For BERT, the long input sequences are disproportionately expensive because attention is quadratic to the sequence[27]. Commonly, the length of the input token sequence is limited to 512. To adapt to our task, for the CCA model, the maximum sequence length is set to 128, while for the sentence longer than 128, it should be truncated to satisfy this setting. For the SCA model, which takes not only values of attribute as input, but also the schema context in relations, the maximum sequence length is set to 256 to contain both contents. Meanwhile, the pre-

Table 4. Statistics of Experimental Datasets

Dataset	Number of Relations	Number of Average Attributes	Number of ASTs	Size of the Training Set	Size of the Validation Set	Size of the Test Set
BMdata[34]	1 361	4	8	3 265	1 089	1 089
Magellan[35]	6 560	8	71	31 016	10 338	10 339
WebTable[36]	38 493	2	83	56 379	18 793	18 793

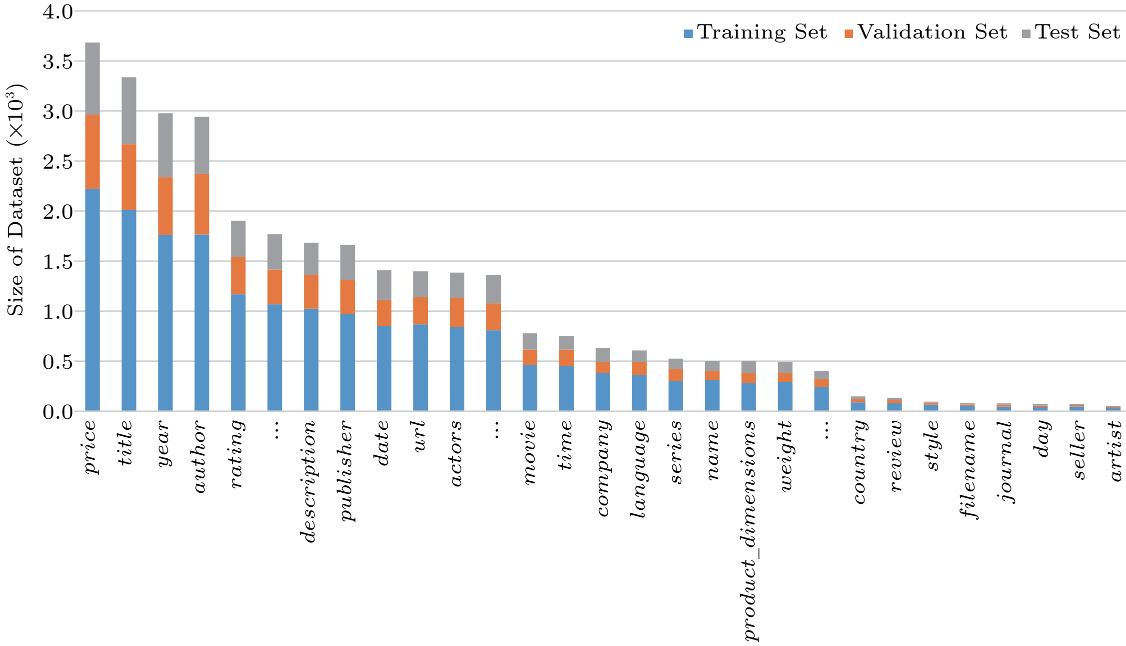


Fig.3. Long-tail data distribution in the Magellan dataset. ··· means some semantic types to present the statistical distribution.

defined sentence separator ([SEP]) is used to separate the sequence pair. With a trick, when the length of the sentence pair is longer than 256, we adopt a simple heuristic truncation approach to truncate the longer sequence one token in turn, effectively avoiding the truncated sequence contains more information of the longer sequence. To summarize, the parameters we set are listed in Table 5.

Table 5. Experimental Parameters Setting

Parameter	Value
Batch size	32
Learning rate (AdamW)	2.0×10^{-5}
Number of epochs	3
Maximum sequence length	128/256

For knowledge base embedding generation, we introduce the DBpedia data[16] and extract the KBVec with Algorithm 1. When retrieving matched entities in KB with keywords, we use lookup service③ and set the maximum number of returned results to 5. Further, with the SPARQL query, we totally get 760 KB ontology classes with parent-child relationships. Therefore, the raw KBVec dimension is set to 760, and each slot represents one specific semantic type. After obtaining the raw KBVec, we introduce PCA for dimensionality reduction and set the number of components to keep to 0.9 which means that the amount of variance that needs to be explained is greater than 90%. For performance comparison, we choose the state-of-the-art model, Sherlock④ as the strong baseline model for AST identification and reproduce the experiment based on different public datasets. Besides Sherlock, we embed attributes with the averaged word vector based on a word2vec model[37] trained by the latest dump of Wikipedia articles and train a basic LR multi-class classification model (AvgWV model) for AST identification.

5.3 Evaluation Metrics

In order to measure the overall performance differences between AST identification models, we calculate macro average F_1 score and weighted average F_1 score respectively based on the test datasets. Different from treating each type equally, the latter weights each type with supports (i.e., the number of samples in each type) and can better evaluate the performance when there are imbalanced type distributions.

Furthermore, the skewed data distributions test the consistency of models for detecting each type accurately. To evaluate whether the model can over-

③Example: Top five related resources which have the keyword of "berlin". http://lookup.dbpedia.org/api/search/KeywordSearch?MaxHits=5&QueryString=berlin, April 2021.

④https://github.com/mitmedialab/sherlock-project, April 2021.

come the impact of long tail distribution on AST identification effectively, Matthews correlation coefficient (MCC) and coefficient of variation (CV) are introduced for detailed per-type evaluation. Among them, MCC[38] is a measure of the quality of classification task frequently used in machine learning. In the multi-class case, MCC can be defined in terms of a confusion matrix $\boldsymbol{M}$ for N types. To simplify the definition, the following intermediate variables are considered.

$$t_n = \sum_i^N \boldsymbol{M}_{in}, \quad p_n = \sum_i^N \boldsymbol{M}_{ni},$$
$$c = \sum_n^N \boldsymbol{M}_{nn}, \quad s = \sum_i^N \sum_j^N \boldsymbol{M}_{ij},$$

where t_n represents the times type n truly occurred, p_n represents the times type n was predicted, c represents the total number of samples correctly predicted, and s represents the total number of samples. Based on above definitions, the multi-class MCC is defined as (9).

$$MCC = \frac{c \times s - \sum_n^N p_n \times t_n}{\sqrt{(s^2 - \sum_n^N p_n^2) \times (s^2 - \sum_n^N t_n^2)}}. \quad (9)$$

The MCC value for the multi-class case ranges in $(-1, 1]$, while the maximum value $+1$ represents a perfect prediction. Compared with other metrics like F_1 score and accuracy, the MCC metric which correctly takes into account the ratio of the confusion matrix size is more informative on imbalanced datasets[39]. Moreover, CV is the ratio of the standard deviation σ to the mean μ, $CV = \sigma/\mu$[40]. We compute CV based on the F_1 score of each type in the test set. The lower the value of CV, the less divergent the per-type F_1 score and the more precise the estimate.

6 Results and Findings

6.1 Effectiveness of Semantic Embedding

With the CCA model and the SCA model, fine-tuning BERT for the AST identification downstream task, we obtain semantic embeddings which are capable of capturing the semantic similarity of attributes, and also obtain well-performed classifiers. We use the performance of the classifier to explore whether semantic embeddings are effective. Table 6 shows the experimental results of different models based on the three datasets. We use the bold font to highlight the highest score for each metric.

The results tell that the basic model, AvgWV, achieves a basic performance for AST identification, since the word2vec model cannot generate embeddings for OOV words, rare words or misspelled words. To fix such problems, BERT utilizes WordPiece for tokenization. With comparison, the fine-tuned semantic embeddings generated by the CCA model and the SCA model perform much better than word2vec. Further, both two models outperform Sherlock, especially the SCA model. For the BM dataset which contains two application domains of bibliography and e-commerce, compared with Sherlock the CCA model brings an improvement of 4.80% for macro average F_1 score and 0.18% for weighted average F_1 score, which verify the effectiveness of semantic embedding obtained based on the pre-trained BERT model. Further, it can be noticed that the introduction of schema context performs comparably with the CCA model, which brings a decrease of macro average F_1 score of 1.37% and a slight improvement of weighted average F_1 score of 0.01%. With detailed analysis, the relations contained in BMdata describe two independent contexts where no semantic overlap exists between attributes from quite different schema contexts. Thus, the features of attribute are enough to distinguish and identify different ASTs. In order to verify the effect of context information for semantic embed-

Table 6. Performance Evaluation for Semantic Embedding

Approach	BMdata		Magellan		WebTable	
	Macro (%)	Weighted (%)	Macro (%)	Weighted (%)	Macro (%)	Weighted (%)
AvgWV	85.88	98.44	70.00	69.46	46.52	75.79
Sherlock[13]	91.90	99.52	90.73	94.02	69.07	88.07
CCA	**96.70**	99.70	93.39	96.30	79.68	93.65
SCA	95.33	**99.71**	**95.52**	**97.92**	**92.16**	**97.34**

Note: The macro average F_1 score and the weighted average F_1 score of different models under different datasets are listed above. Among them, AvgWV is the basic model we construct, which embeds the attribute values with the averaged word vector and uses LR for classification. Sherlock is a strong baseline model.

ding, we introduce Magellan data, a multi-domain relational data with richer topics on the basis of BMdata. As introduced in Subsection 5.1, the Magellan dataset involves nine domains of data (e.g., animal, book, and music) and the ASTs need to be restricted and identified with the help of specific context. The SCA model outperforms Sherlock, achieving 95.52% vs 90.73% macro average F_1 score and 97.92% vs 94.02% weighted average F_1 score. Except for Magellan data, the high-quality relational data which is mainly used for entity resolution, we also adopt web table corpus. With the preliminary consideration of attribute-wise features, the baseline Sherlock model performs poorly on this relatively low-quality web data which covers various domains. However, our SCA model can distinguish and identify features between different ASTs by combining schema context and achieve a 23.09% gain in macro average F_1 score and a 9.27% gain in weighted average F_1 score.

In general, both the CCA model and the SCA model can generate distinguished and representative embeddings for further classification. Especially, for datasets with mixed domains where attribute types cannot be determined by single column content, the SCA model will play an important role for semantic embedding generation.

6.2 Effectiveness of Knowledge Base Embedding

By retrieving matched entities and corresponding ontology classes in DBpedia, we obtain KBVec which represents the potential type characteristics of attributes. To verify the effectiveness of KBVec, we concatenate it with the semantic embedding generated from different approaches and train classifiers with LR for further performance evaluation.

Experimental results in Table 7 demonstrate that the introduction of KBVec can effectively improve the effect of AST identification. Especially for the basic classifier representing the attributes with the averaged word vector (AvgWV), the macro average F_1 score and the weighted average F_1 score are increased by 8.74% and 12.71% respectively based on the Magellan dataset. Moreover, to reduce the impact of high-dimensional sparse KBVec on classification, we adopt PCA to transform the raw 760-dimensional KBVec linearly into a low-dimensional space. The results tell that ensembling semantic embedding and the compressed KBVec as input features of the LR classifier can obtain the best classification effect. With a few exceptions, individual results show that the compressed KBVec performs worse than the raw 760-dimensional KBVec. For example, for CCA semantic embeddings of dataset Magellan, the introduction of PCA causes a decrease of the macro average F_1 score of 0.05%. With analysis, a certain amount of information is lost induced by dimensionality reduction inevitably, leading to the reduction in performance. Considering that the introduction of PCA reduces the computational complexity and improves the execution efficiency, it is necessary to find the balance between execution effect and efficiency.

In general, the generated KBVec supplements the semantic embedding and enhances the quality of embedding. On the other hand, when the semantics to be identified are un-predefined, KBVec can be used to provide candidate types to help semantic inference (Subsection 6.6). In addition, the adoption of PCA for dimensionality reduction can speed up modeling and overcome the impact of sparse high-dimensional features.

Table 7. Performance Evaluation for Knowledge Base Embedding

Semantic Embedding Approach	KBVec	PCA	BMdata		Magellan		WebTable	
			Macro (%)	Weighted (%)	Macro (%)	Weighted (%)	Macro (%)	Weighted (%)
AvgWV	/	/	85.88	98.44	70.00	69.46	46.52	75.79
	760	/	88.86	**98.82**	78.74	82.17	**54.43**	**81.18**
	760 (29/37/38)	0.9	**89.52**	98.73	**78.91**	**82.41**	53.85	80.87
CCA	/	/	96.70	99.70	93.39	96.30	79.68	93.65
	760	/	95.33	99.71	**94.37**	96.58	80.25	93.74
	760 (29/37/38)	0.9	**97.45**	**99.80**	94.32	**96.63**	**80.84**	**93.79**
SCA	/	/	**95.33**	**99.71**	95.52	97.92	92.16	97.34
	760	/	95.33	99.71	96.03	98.01	92.99	97.55
	760 (29/37/38)	0.9	95.01	99.70	**96.20**	**98.04**	**94.24**	**97.63**

Note: With ablation experiments, we explore the effectiveness of knowledge base embedding. For each approach, the first row refers to that the semantic embeddings are concatenated with no KBVec, while the second row refers to that the semantic embeddings are concatenated with the raw 760-dimensional KBVecs and the third row refers to that the semantic embeddings are concatenated with the compressed KBVecs, the numbers in brackets indicate the dimension after compression and the decimal means the minimum amount of variance that needs to be explained with PCA.

6.3 Effectiveness of Ensemble Approaches

With comprehensive analysis, the semantic embedding generated by the SCA model and the knowledge embedding generated by our extraction algorithm characterize attributes in relations well. In Subsection 6.2, the fine-tuned semantic embeddings of attributes are concatenated with knowledge base embeddings as input to train a multi-class classifier corresponding to the Feature Ensemble with LR mentioned in Subsection 4.4. We explore other ensemble approaches in the SCA-KB model in this subsection.

In Table 8, Score Ensemble is to determine the prediction results with the averaged probabilities of the SCA model and the basic model where the averaged word vector of attributes is concatenated with KBVec. Feature Ensemble with BERT concatenates the compressed KBVec with BERT representation to fine-tune the model for the downstream classification task. Feature Ensemble with LR is the approach used in Subsection 6.2 and the best results in Table 7 are listed. Experimental results show that for the BM dataset, both Score Ensemble and Feature Ensemble with LR perform well. For the Magellan dataset and Web dataset, only feature ensemble with LR achieves good performance, e.g., compared with the SCA model based on the Magellan dataset, it obtains the macro average F_1 score of 96.20% higher than 95.52% and the weighted average F_1 score of 98.04% higher than 97.92%. At the same time, it can be noted that Feature Ensemble with BERT performs poorly on three datasets. One potential reason is that too many parameters in the neural network lead to over-fitting with high variance.

In general, for attributes easy to distinguish, Score Ensemble and Feature Ensemble with LR can be used for semantic identification. While for the general data, Feature Ensemble with LR ensembles the fine-tuned semantic embedding and knowledge base embedding effectively and improves the identification performance.

6.4 Per-Type Evaluation

Taking the Magellan data as an example and taking the per-type F_1 score of Sherlock as the baseline, Fig.4 shows differences between models. The SCA model outperforms Sherlock in 46 out of 71 semantic types while underperforming Sherlock in four semantic types. Among them, the F_1 score of *artist*, *asin*, *month*, *review_count* and so on is significantly higher than that of Sherlock. In more details, as seen in Table 9, samples with the ground truth label of *artist* are misidentified as *restaurant*, *city*, *director*, *company*, and *category* in Sherlock. While our SCA model greatly reduces such errors, e.g., the F_1 score of *artist* is improved from 48.00% to 89.66%. Moreover, introducing knowledge base embedding improves the performance for 47 semantic types out of 71 semantic types with four types getting worse and 20 types getting equal. Among them, on the basis of SCA semantic embedding, the introduction of external knowledge especially improves the identification performances for *album*, *song*, *width* and so on. For example, in the SCA model, *album* is wrongly identified as *price*, *genres*, *song* and *copyright* with the F_1 score of 76.92%, while in Feature Ensemble with LR, *album* is only confused with *copyright* or *price* with the F_1 score of 89.66%. More examples can be seen in Table 9.

Our SCA model generates semantic embeddings by considering the context in relations which compensate the lower-expressive abilities of numerical (e.g., *asin*, *month*, and *review_count*) and confusing types (e.g., *artist*, *director*, and *authors*). Further, taking the fine-tuned semantic embedding and the compressed knowledge base embedding into consideration, our Feature Ensemble with the LR model can improve or maintain the identification performances of most ASTs. The knowledge base embeddings extracted by retrieving DBpedia featurize the potential type characteristics of attributes, which are not available by semantic embeddings.

Table 8. Performance Evaluation for Ensemble Approaches

Approach	BMdata		Magellan		WebTable	
	Macro (%)	Weighted (%)	Macro (%)	Weighted (%)	Macro (%)	Weighted (%)
Sherlock[13]	91.90	99.52	90.73	94.02	69.07	88.07
SCA	95.33	99.71	95.52	97.92	92.16	97.34
Score ensemble	**<u>98.04</u>**	<u>99.72</u>	95.35	97.72	91.21	96.83
Feature ensemble with BERT	86.88	98.96	94.24	97.11	88.87	95.84
Feature ensemble with LR	<u>97.45</u>	**<u>99.80</u>**	**<u>96.20</u>**	**<u>98.04</u>**	**<u>94.24</u>**	**<u>97.63</u>**

Note: The better the performance is, the more effective the ensemble approach will be. The underline highlights the performance better than that of the SCA model and the bold highlights the highest score for each metric in each specific dataset.

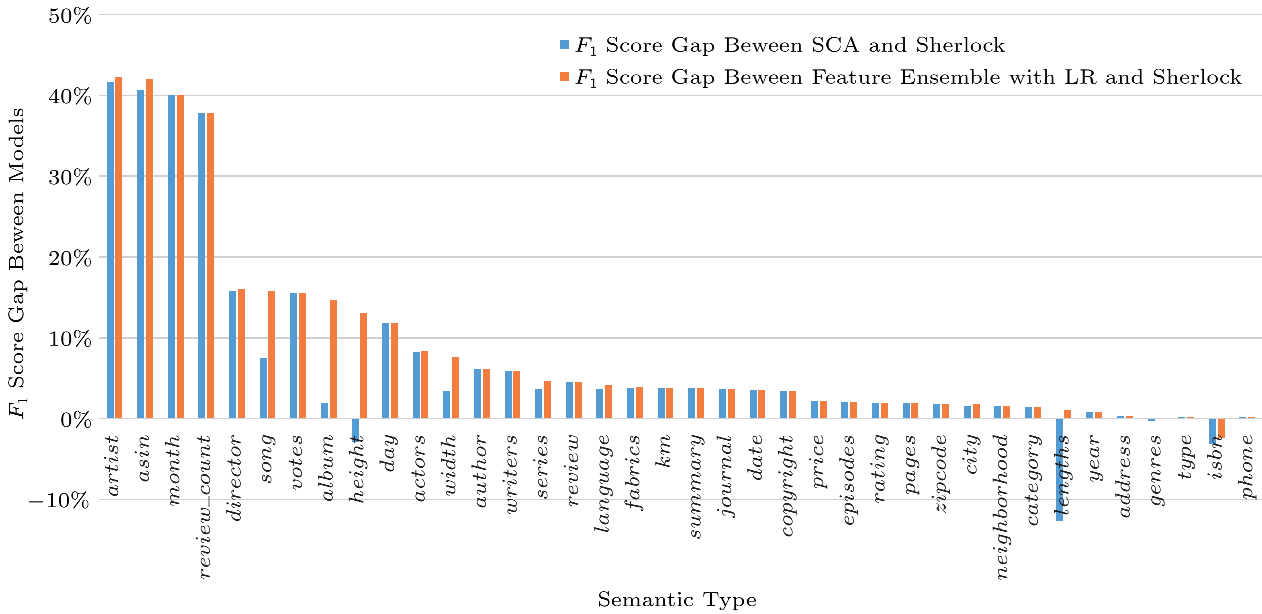


Fig.4. Part of F_1 score gaps between two models. Taking the per-type F_1 score of Sherlock as the baseline, the blue histograms represent the performance differences between SCA and Sherlock, while the orange histograms represent the performance differences between feature ensemble with LR and Sherlock. The gap greater than 0 indicates the performance better than that of Sherlock, but not vice versa.

Table 9. Examples for Semantic Type Identification

Ground Truth	Sherlock		SCA		Feature Ensemble with LR	
	Prediction	F_1 Score (%)	Prediction	F_1 Score (%)	Prediction	F_1 Score (%)
width	*review_count, color, height,* ***width****, length*	46.04	***width****, length, height, weight*	49.46	***width****, length, height, weight*	53.69
album	*title, author, company, summary, artist,* ***album****, copyright*	75.00	*price, genres, song,* ***album****, copyright*	76.92	***album****, copyright, price*	89.66
artist	*restaurant, city, director, company, category,* ***artist***	48.00	*director,* ***artist****, album*	89.66	***artist****, album*	90.32
actors	*author,* ***actors****, director, creators*	91.42	***actors****, director*	99.60	***actors***	99.80
song	*title,* ***song***	84.21	***song***	91.67	***song***	100.00

Note: The table lists the prediction results (where the bold font indicates true prediction and the others are false prediction) and F_1 scores for samples with the specific ground truth labels. It should be noted that for types with only true prediction results and F_1 score less than 100%, all samples with the ground truth label of the specific type are identified correctly (i.e., the recall is 100%), but there are still other samples that are identified as the specific type falsely.

6.5 Long-Tail Types Identification

The experimental datasets we used show clear long tail distributions, where the common ASTs appear with high frequency, while some ASTs are less-represented with few training samples. With detailed analysis of the evaluation results in Table 8, we observe that the improvements of macro average F_1 score are generally higher than those of weighted average F_1 score, which suggests that the significant improvements come from boosting the accuracy for the less-represented types.

Our model tends to perform well in generalizing to all ASTs especially the low-frequency types that comprise the long-tail. To evaluate the performance for long-tail type identification, we calculate MCC on the overall prediction results and CV based on the F_1 score for each type. The experimental results listed in Table 10 demonstrate that the ensemble model effectively improves MCC with schema context and external knowledge introduced. A higher MCC value close to 1 indicates that the experimental result is a perfect prediction compared with the ground truth labels generated from the column headers. Besides a lower CV indicates that the predicted F_1 score of each category is less variant, i.e., our model has stable performances on long-tail datasets.

Further, taking Magellan data as an example, the

Table 10. *MCC* and *CV* for Different Approaches

Approach	BMdata		Magellan		WebTable	
	MCC	*CV*	*MCC*	*CV*	*MCC*	*CV*
AvgWV	0.983	0.406	0.737	0.561	0.784	0.922
Sherlock[13]	0.993	0.137	0.937	0.170	0.871	0.404
SCA	0.996	0.078	0.979	0.132	0.973	0.202
Feature ensemble with LR	**0.998**	**0.072**	**0.981**	**0.113**	**0.977**	**0.130**

Note: The table lists *MCC* and *CV* for different approaches. The bold font is used to highlight the highest score for each metric.

normalized confusion matrices of Sherlock and Feature Ensemble with LR are shown in Fig.5 and Fig.6 respectively, where the vertical axis represents the true types of attributes which are sorted by the support of each type in the training set in descending order, and the horizontal axis indicates the prediction types. The darker the color is, the closer the value in the matrix is to 1, i.e., more attributes are predicted as the type corresponding to the horizontal axis. It can be clearly seen that values along the diagonal of the matrix are mostly close to 1, which means that both the baseline Sherlock model and the feature ensemble model can predict most ASTs accurately. With detailed comparison, there are partial non-zero values in the lower left corner of the matrix in Fig.5, that is, for the long-tail types with fewer samples, the Sherlock model predicts them as high-frequency types incorrectly. While in Fig.6, such false prediction in-

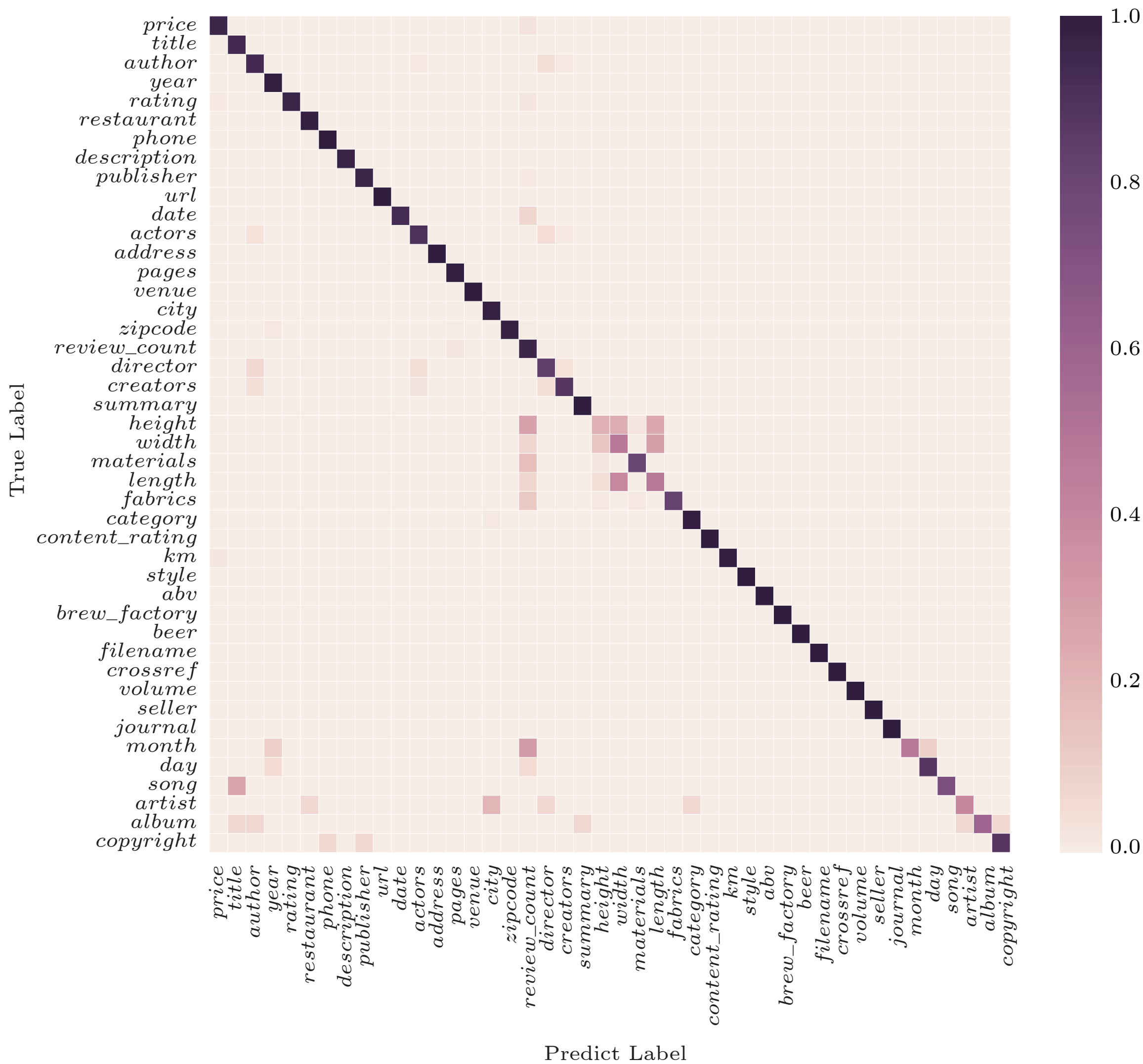


Fig.5. Normalized confusion matrix of ASTs prediction with Sherlock.

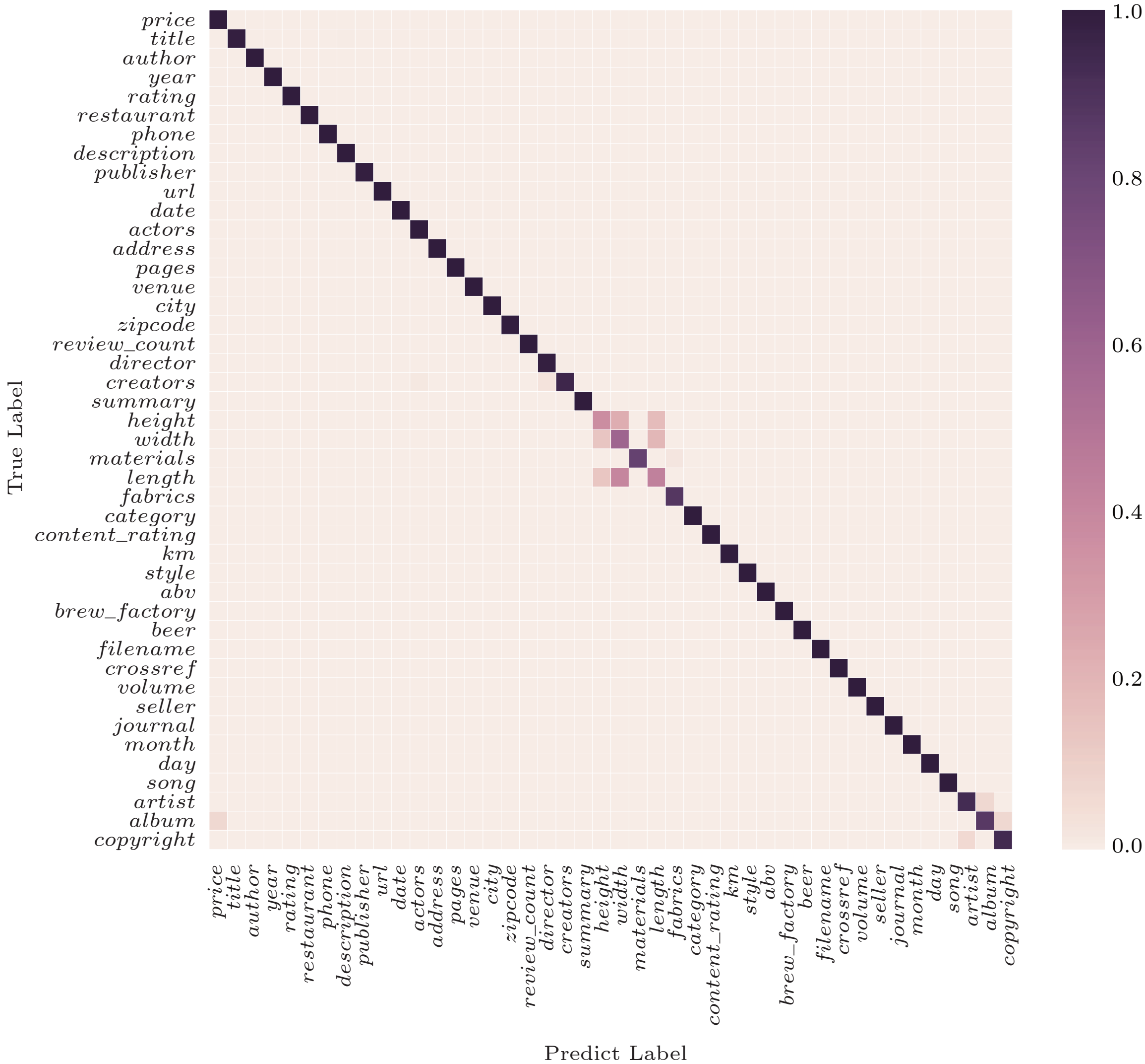


Fig.6. Normalized confusion matrix of ASTs prediction with Feature Ensemble with LR.

stances are significantly reduced, which further demonstrates that incorporating context and external knowledge can effectively alleviate the problem of lacking training data for rare types, and confirms that our model can effectively boost the accuracy for the long-tail types.

6.6 Case Study: Unknown Type Identification

We transform AST identification for relational attributes into multi-class classification with the assumption that all the semantic types to be identified are predefined. For unknown types which do not exist in the training set, the well-trained classifier just returns the relative most likely matching semantics. In order to provide more accurate candidate types for such samples and enhance the robustness of our model, we conduct the unknown type detection according to the maximum predicted probability given by the classifier and provide candidate classes with reference to the extracted raw KBVec.

With the Efthymiou[41] data used in [9] and the SCA model trained based on the Magellan data, we conduct preliminary experiments for unknown AST identification. Compared with the predefined types in the Magellan dataset, Efthymiou contains both predefined and un-predefined ASTs. In our ensemble model, we regard the samples with low prediction probabilities which are less than the threshold we define (in this case study, we set the threshold to 0.5) as anomalous samples with unknown types. Further, AST identification for anomalous samples is performed by referring to the raw KBVecs and the candidate types corresponding to the non-zero slots in KBVecs are highlighted. Table 11 shows examples for the

Table 11. Case Study for Samples with Unknown Types

Sample ID	Attribute Value	Ground Truth	False Prediction	Candidate Class
823	"Gimpo International Airport", "Gimhae International Airport", "Cheongju International Airport", "Gwangju Airport", "Daegu International Airport"	*airport*	*city*	***airport***, *architectural structure*, *infrastructure*, *place*
1067	"Lesser yellowlegs", "Willet", "Wandering tattler", "Spotted sandpiper", "Whimbrel"	*bird*	*color*	*aircraft*, *animal*, ***bird***, *species*
1242	"Al-Jamiatul Ahlia Darul Ulum Moinul Islam", "Al-Jamiah Al-Islamiah Patiya", "Al-Jamiatul Arabiatul Islamiah, Ziri", "Jamia Tawakkulia Renga Madrasah", "Jamiah Islamiah Yunusia Brahmanbaria"	*university*	*company*	*agent*, *educational institution*, *organisation*, ***university***
1557	"Carnegie Library of Homestead, Munhall, Pennsylvania", "606 W. Cork St, Winchester, Virginia", "Box 360, Winchester, Virginia (Frank Reed Horton's mailbox)", "Ivanhoe Club Building, 3215 Park Avenue, Kansas City, Missouri", "410-11 Land Bank Building, Kansas City 6, Missouri"	*building*	*neighborhood*	*place*, *educational institution*, *organisation*, ***building***

Note: In the table, false prediction is the prediction results of our SCA model, while the candidate classes are obtained by our SCA-KB model, Feature Ensemble with LR. In detail, the candidate classes inferred from KBVec are sorted in descending order of the probability values.

case study and four candidate classes are listed for each sample. For example, for anomalous sample 823 with the ground truth label of *airport*, the SCA model misidentifies it as *city* since attribute values of *airport* contain confusing city names. In our ensemble model, the true label *airport* is highlighted by referring to the KBVec.

7 Conclusions

In this paper, we presented a context-aware method to solve the AST identification problem. The main idea of this method is to transform the AST identification into a multi-class classification problem. For this purpose, we proposed an SCA model which is able to generate the embedding for each attribute and figure out the AST for the attribute based on the embedding. Our special design made the embedding capture the semantics of the target attributes, as well as the related schema context. Since the actual AST of the attribute might not be included in the training set, which will lead to that the SCA model could not figure out the semantic types properly, we proposed an SCA-KB model which enhances the embeddings of attributes by introducing the entities and ontology types in the knowledge base. For this reason, we proposed an SCA-KB model which enhances the embeddings of attributes by introducing the entities and ontology types in the knowledge base. Compared with the SCA model, the SCA-KB model could figure out the semantic types of attributes that are included in the knowledge base but not included in the training set. We conducted extensive experiments and the results demonstrated that our context-aware method outperformed the state-of-the-art approaches by a large margin, up to 6.14% and 25.17% in terms of macro average F_1 score, and up to 0.28% and 9.56% in terms of weighted average F_1 score over high-quality and low-quality datasets respectively.

For future research, we will further explore improvements of our AST identification method from two perspectives. On the one hand, the ontology types in the knowledge base are hierarchical, while in our SCA-KB model, the ontology types are used independently in a flat manner. To fill in this gap, we will attempt to enhance the embeddings of attributes in our SCA-KB model by preserving the hierarchical context of ontology types. On the other hand, we will try to improve the efficiency of our context-aware method. In the current design, both the SCA and SCA-KB model are based on BERT, which has been well recognized as a prohibitively expensive pre-trained model. To improve the efficiency, we will attempt to optimize the used pre-trained model by introducing weight quantization[42], knowledge distillation[43] and parameter sharing[44], which are orthogonal to our work.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] Kandel S, Paepcke A, Hellerstein J, Heer J. Wrangler: Interactive visual specification of data transformation

scripts. In *Proc. the 2011 SIGCHI Conference on Human Factors in Computing Systems*, May 2011, pp.3363–3372. DOI: 10.1145/1978942.1979444.

[2] Rahm E, Bernstein P A. A survey of approaches to automatic schema matching. *The VLDB Journal*, 2001, 10(4): 334–350. DOI: 10.1007/s007780100057.

[3] Zapilko B, Zloch M, Schaible J. Utilizing regular expressions for instance-based schema matching. In *Proc. the 7th International Conference on Ontology Matching*, Nov. 2012, pp.240–241. DOI: 10.5555/2887596.2887623.

[4] Venetis P, Halevy A, Madhavan J, Paşca M, Shen W, Wu F, Miao G X, Wu C. Recovering semantics of tables on the web. *Proceedings of the VLDB Endowment*, 2011, 4(9): 528–538. DOI: 10.14778/2002938.2002939.

[5] Snipes G. Google data studio. *Journal of Librarianship and Scholarly Communication*, 2018, 6(1): eP2214. DOI: 10.7710/2162-3309.2214.

[6] Kaelin M. Microsoft power BI: A cheat sheet. Technical Report, Techrepublic, 2019. https://www.techrepublic.com/article/microsoft-power-bi-a-smart-persons-guide, July 2023.

[7] Black D. Data wrangling 'decoder ring' homogenizes polyglot data lakes. Technical Report, Enterprise Tech., 2016. https://www.enterpriseai.news/2016/02/11/trifactas-data-wrangling-decoder-ring-homogenizes-polyglot-data-lakes/, July 2023.

[8] Zhao C, He Y Y. Auto-EM: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning. In *Proc. the 2019 World Wide Web Conference*, May 2019, pp.2413–2424. DOI: 10.1145/3308558.3313578.

[9] Chen J Y, Jiménez-Ruiz E, Horrocks I, Sutton C. Learning semantic annotations for tabular data. In *Proc. the 28th International Joint Conference on Artificial Intelligence*, Jul. 2019, pp.2088–2094. DOI: 10.24963/ijcai.2019/289.

[10] Chen J Y, Jiménez-Ruiz E, Horrocks I, Sutton C. ColNet: Embedding the semantics of web tables for column type prediction. In *Proc. the 33rd AAAI Conference on Artificial Intelligence*, Feb. 2019, pp.29–36. DOI: 10.1609/aaai.v33i01.330129.

[11] Ramnandan S K, Mittal A, Knoblock C A, Szekely P. Assigning semantic labels to data sources. In *Proc. the 12th European Semantic Web Conference*, May 31–June 4, 2015, pp.403–417. DOI: 10.1007/978-3-319-18818-8_25.

[12] Pham M, Alse S, Knoblock C A, Szekely P. Semantic labeling: A domain-independent approach. In *Proc. the 15th International Semantic Web Conference*, Oct. 2016, pp.446–462. DOI: 10.1007/978-3-319-46523-4_27.

[13] Hulsebos M, Hu K, Bakker M, Zgraggen E, Satyanarayan A, Kraska T, Demiralp Ç, Hidalgo C. Sherlock: A deep learning approach to semantic data type detection. In *Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Jul. 2019, pp.1500–1508. DOI: 10.1145/3292500.3330993.

[14] Krishna S. Introduction to Database and Knowledge-Base Systems. World Scientific Publishing, 1992. DOI: 10.1142/1374.

[15] Gao Y, Liang J, Han B, Yakout M, Mohamed A. Building a large-scale accurate and fresh knowledge graph. In *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2018. https://kdd2018tutorialt39.azurewebsites.net/, July 2023.

[16] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A nucleus for a web of open data. In *Proc. the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference*, Nov. 2007, pp.722–735. DOI: 10.1007/978-3-540-76298-0_52.

[17] Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proc. the 2008 ACM SIGMOD International Conference on Management of Data*, Jun. 2008, pp.1247–1250. DOI: 10.1145/1376616.1376746.

[18] Rebele T, Suchanek F, Hoffart J, Biega J, Kuzey E, Weikum G. YAGO: A multilingual knowledge base from Wikipedia, Wordnet, and Geonames. In *Proc. the 15th International Semantic Web Conference*, Oct. 2016, pp.177–185. DOI: 10.1007/978-3-319-46547-0_19.

[19] Zwicklbauer S, Einsiedler C, Granitzer, M, Seifert C. Towards disambiguating Web tables. In *Proc. the 2013 International Semantic Web Conference*, Oct. 2013, pp.205–208.

[20] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press, 2016.

[21] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, 521(7553): 436–444. DOI: 10.1038/nature14539.

[22] Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*, 2015, 61: 85–117. DOI: 10.1016/j.neunet.2014.09.003.

[23] Wang W, Zhang M H, Chen G, Jagadish H V, Ooi B C, Tan K L. Database meets deep learning: Challenges and opportunities. *ACM SIGMOD Record*, 2016, 45(2): 17–22. DOI: 10.1145/3003665.3003669.

[24] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In *Proc. the 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp.6000–6010. DOI: 10.5555/3295222.3295349.

[25] He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In *Proc. the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2016, pp.770–778. DOI: 10.1109/CVPR.2016.90.

[26] Ba J L, Kiros J R, Hinton G E. Layer normalization. arXiv: 1607.06450, 2016. https://arxiv.org/abs/1607.06450, July 2023.

[27] Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv: 1810.04805, 2018. https://arxiv.org/abs/1810.04805, July 2023.

[28] May C, Wang A, Bordia S, Bowman S R, Rudinger R.

On measuring social biases in sentence encoders. arXiv: 1903.10561, 2019. https://arxiv.org/abs/1903.10561, July 2023.

[29] Qiao Y F, Xiong C Y, Liu Z H, Liu Z Y. Understanding the behaviors of BERT in ranking. arXiv: 1904.07531, 2019. https://arxiv.org/abs/1904.07531, July 2023.

[30] Harris Z S. Distributional structure. *Word*, 1954, 10(2/3): 146–162. DOI: 10.1080/00437956.1954.11659520.

[31] Erk K. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 2012, 6(10): 635–653. DOI: 10.1002/lnco.362.

[32] Pennington J, Socher R, Manning C. GloVe: Global vectors for word representation. In *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing*, Oct. 2014, pp.1532–1543. DOI: 10.3115/v1/D14-1162.

[33] Le Q, Mikolov T. Distributed representations of sentences and documents. In *Proc. the 31st International Conference on Machine Learning*, Jun. 2014, pp.1188–1196. DOI: 10.5555/3044805.3045025.

[34] Köpcke H, Thor A, Rahm E. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 2010, 3(1/2): 484–493. DOI: 10.14778/1920841.1920904.

[35] Konda P, Das S, Suganthan G C P, Doan A, Ardalan A, Ballard J R, Li H, Panahi F, Zhang H J, Naughton J, Prasad S, Krishnan G, Deep R, Raghavendra V. Magellan: Toward building entity matching management systems. *Proceedings of the VLDB Endowment*, 2016, 9(12): 1197–1208. DOI: 10.14778/2994509.2994535.

[36] Eberius J, Braunschweig K, Hentsch M, Thiele M, Ahmadov A, Lehner W. Building the Dresden Web Table Corpus: A classification approach. In *Proc. the 2nd International Symposium on Big Data Computing*, Dec. 2015, pp.41–50. DOI: 10.1109/BDC.2015.30.

[37] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv: 1301.3781, 2013. https://arxiv.org/abs/1301.3781, July 2023.

[38] Matthews B W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 1975, 405(2):442–451. DOI: 10.1016/0005-2795(75)90109-9.

[39] Chicco D. Ten quick tips for machine learning in computational biology. *BioData Mining*, 2017, 10(1): Article No. 35. DOI: 10.1186/s13040-017-0155-3.

[40] Everitt B S. The Cambridge Dictionary of Statistics (2nd edition). Cambridge University Press, 2002.

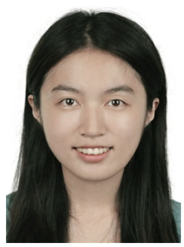
[41] Efthymiou V, Hassanzadeh O, Rodriguez-Muro M, Christophides V. Matching Web tables with knowledge base entities: From entity lookups to entity embeddings. In *Proc. the 16th International Semantic Web Conference*, Oct. 2017, pp.260–270. DOI: 10.1007/978-3-319-68288-4_16.

[42] Shen S, Dong Z, Ye J Y, Ma L J, Yao Z W, Gholami A, Mahoney M W, Keutzer K. Q-BERT: Hessian based ultra low precision quantization of BERT. In *Proc. the 34th AAAI Conference on Artificial Intelligence*, Feb. 2020, pp.8815–8821. DOI: 10.1609/aaai.v34i05.6409.

[43] Jiao X Q, Yin Y C, Shang L F, Jiang X, Chen X, Li L L, Wang F, Liu Q. TinyBERT: Distilling BERT for natural language understanding. arXiv: 1909.10351, 2019. https://arxiv.org/abs/1909.10351, July 2023.

[44] Lan Z Z, Chen M D, Goodman S, Gimpel K, Sharma P, Soricut R. ALBERT: A lite BERT for self-supervised learning of language representations. arXiv: 1909.11942, 2019. https://arxiv.org/abs/1909.11942, July 2023.

Yue Ding received her B.S. degree in Internet of Things from Nanjing University of Posts and Telecommunications, Nanjing, in 2018. She is currently pursuing her M.S. degree in Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education and School of Information at Renmin University of China, Beijing. Her research interests include data integration and machine learning.

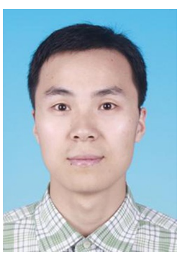


Yu-He Guo received her B.S. degree in computer science from Renmin University of China, Beijing, in 2018. She is currently pursuing her M.S. degree in Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education and School of Information in Renmin University of China, Beijing. Her research interests lie in natural language processing and data integration.

Wei Lu is currently an associate professor in Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education and School of Information at Renmin University of China, Beijing. He received his Ph.D. degree in computer applied technology from Renmin University of China, Beijing, in 2011. His research interests include query processing in the context of spatiotemporal, cloud database systems and applications. He is a member of CCF.

Hai-Xiang Li is currently a senior expert at Tencent (Beijing) Technology Company Limited, Beijing. His research interests include transaction processing, query optimization, distributed consistency, high availability, database system architecture, cloud database and distributed database systems. He is a member of CCF.

Mei-Hui Zhang received her Ph.D. degree in computer science from National University of Singapore, Singapore, in 2013. She is currently a professor with Beijing Institute of Technology, Beijing, and was an assistant professor with Singapore University of Technology and Design, Singapore, from 2014 to 2017. Her research interests include big data management and analytics, large-scale data integration, modern database systems, block chain and AI. She has served as PC Vice-Chair of ICDE 2018 and associate editor of VLDB 2018, VLDB 2019, VLDB 2020 and SIGMOD 2021. She is a winner of VLDB 2020 Early Career Research Contribution Award. She is a member of CCF, ACM and IEEE.



Hui Li is currently a professor in College of Computer Science and Technology at Guizhou University, Guiyang. He received his Ph.D. degree in computer software and theory from Renmin University of China, Beijing, in 2012. His research interests include large-scale data analytics, high-performance database systems and data-driven intelligent applications. He is a member of CCF, ACM and IEEE.

An-Qun Pan is a technical director at Tencent (Shenzhen) Technology Company Limited, Shenzhen, with more than 15 years of experience in the research and development of distributed computing and storage systems. He is currently responsible for the research and development of distributed database system (TDSQL). He is a member of CCF.

Xiao-Yong Du is a professor in Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education and School of Information at Renmin University of China, Beijing. He received his Ph.D. degree in computer science from Nagoya Institute of Technology, Nagoya, in 1997. His research focuses on intelligent information retrieval, high performance database and unstructured data management. He is a fellow of CCF.