AnaMeta: A Table Understanding Dataset of Field Metadata Knowledge Shared by Multi-dimensional Data Analysis Tasks

Xinyi He[1*] **Mengyu Zhou**[2†] **Mingjie Zhou**[3*] **Jialiang Xu**[4*] **Xiao Lv**[2]
Tianle Li[5*] **Yijia Shao**[6*] **Shi Han**[2] **Zejian Yuan**[1] **Dongmei Zhang**[2]
[1] Xi'an Jiaotong University [2] Microsoft Research [3] The University of Hong Kong
[4] University of Illinois at Urbana-Champaign [5] University of Waterloo [6] Peking University
hxyhxy@stu.xjtu.edu.cn, mjzhou@connect.hku.hk, jx17@illinois.edu,
t29li@uwaterloo.ca, shaoyj@pku.edu.cn, yuan.ze.jian@xjtu.edu.cn,
{mezho, xilv, shihan, dongmeiz}@microsoft.com

Abstract

Tabular data analysis is performed every day across various domains. It requires an accurate understanding of field semantics to correctly operate on table fields and find common patterns in daily analysis. In this paper, we introduce the AnaMeta dataset, a collection of 467k tables with derived supervision labels for four types of commonly used field metadata: measure/dimension dichotomy, common field roles, semantic field type, and default aggregation function. We evaluate a wide range of models for inferring metadata as the benchmark. We also propose a multi-encoder framework, called KDF, which improves the metadata understanding capability of tabular models by incorporating distribution and knowledge information. Furthermore, we propose four interfaces for incorporating field metadata into downstream analysis tasks.

1 Introduction

Tabular data analysis is performed every day in popular tools such as Excel, Google Sheets, and Tableau (Rebman Jr et al., 2022) for a wide range of domains including education, research, engineering, finance, HR, *etc.* To help non-expert users, various machine learning tasks are proposed to automate and accelerate the analysis process. For example, TableQA & Text2SQL (Dong and Lapata, 2016; Katsogiannis-Meimarakis and Koutrika, 2021), analysis & visualization recommendations (Zhou et al., 2021; Wu et al., 2021), insights mining (Ding et al., 2019; Law et al., 2020), *etc.*

These analysis tasks require an accurate understanding of field semantics to correctly operate on table fields (or columns) and to further find common patterns in daily analysis. Such *analysis knowledge* of field semantics is often shared across multiple tasks. In real-world applications, we also call it **field metadata** in contrast to the raw tabular input which does not directly provide this information.

For example, **measure / dimension** dichotomy is one such metadata used in Tableau (Hoelscher and Mortimer, 2018) and Excel (Ding et al., 2019) across diverse features. Its definition is inspired by dimensional modeling in databases (Golfarelli et al., 1998; Kimball and Ross, 2013). It involves categorizing each field in a table as either measure or dimension. A measure (field) contains numerical measurement results on which calculations can be made, such as sum, count, average, minimum, and maximum. *E.g.*, "Price" and "Discount" fields of Table 1 are measures. On the other hand, a dimension (field) contains categorical information and can be used for filtering, grouping, and labeling. *E.g.*, "Product Name" and "Category" fields of Table 1 are dimensions. Infeasible analysis might be generated based on incorrect classification of measures and dimensions in a table, because the feasible operations for measure and dimension are largely different. *E.g.*, for Table 1, it is impossible to draw a bar chart without measures – only mapping dimension "Product Name" to x-axis and dimension "Product Id" to y-axis.

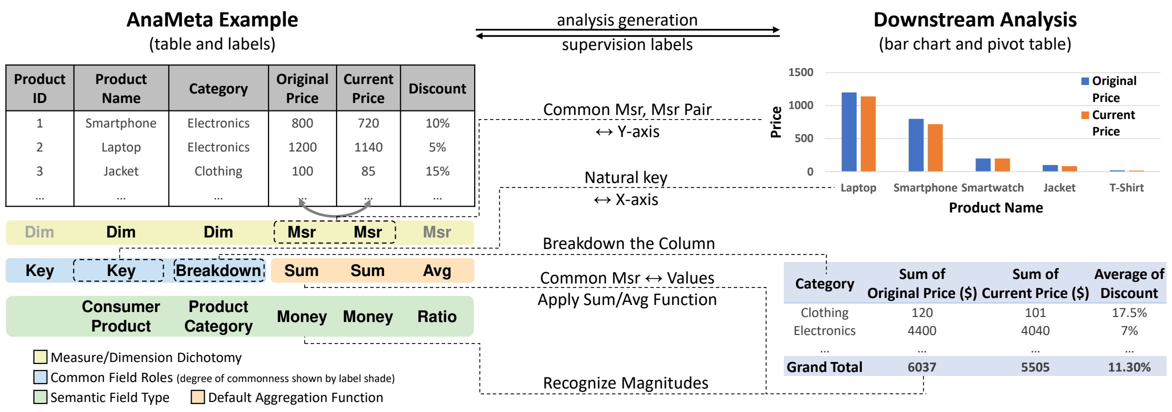
Beyond the simple dichotomy of measure / dimension, what are other types of useful field metadata? For each type of metadata, where do we find labels to evaluate or train models? To address these questions, in this paper we define the following four types of commonly used field metadata, and collect the **AnaMeta** (**Ana**lysis **Meta**data) dataset incorporating tables from 3 diverse sources with derived supervision labels of the metadata:

§2.1 *Measure/dimension dichotomy*: Categorizing each field in a table as a measure or dimension.

§2.2 *Common field roles*: Identifying whether a measure field is commonly used as an analysis target, whether a dimension field is a natural key or

[*] The contributions by Xinyi He, Mingjie Zhou, Jialiang Xu, Tianle Li and Yijia Shao have been conducted and completed during their internships at Microsoft Research Asia, Beijing, China.

[†] Corresponding author.



Product ID	Product Name	Category	Original Price	Current Price	Discount
1	Smartphone	Electronics	800	720	10%
2	Laptop	Electronics	1200	1140	5%
3	Jacket	Clothing	100	85	15%
...	...	...	...	...	...

Category	Sum of Original Price ($)	Sum of Current Price ($)	Average of Discount
Clothing	120	101	17.5%
Electronics	4400	4040	7%
...	...	...	...
Grand Total	6037	5505	11.30%

Figure 1: An Example Table of AnaMeta dataset with Field Metadata Task Supervision and Downstream Analysis.

commonly used for breakdown operation.

§2.3 *Semantic field type*: Identifying types of measure fields and dimension fields from our established measure type taxonomy (*e.g.*, "Money", "Ratio" in Figure 1) and a dimension type taxonomy provided by a knowledge graph (*e.g.*, "Consumer Product" in Figure 1), and determining whether two measures are comparable with each other.

§2.4 *Default aggregation function*: Identifying the most appropriate default aggregation function for a measure field. (*e.g.*, "Avg", "Sum"in Figure 1)

The collection of the AnaMeta dataset is a complex process that requires a significant amount of data and supervision. Due to the limitations of obtaining this information directly from raw tables, we have employed a multi-source approach to collecting AnaMeta, which includes spreadsheet datasets (45,361 tables), public web table datasets (404,152 tables), and public synthetic datasets (17,995 tables). Supervision is smartly collected from downstream tasks, manual labels through crowdsourcing, and existing information attached to tables(§3.2). After preprocessing, AnaMeta contains 152,092 fields with measure/dimension, 149,197 with common field roles, 1,730,494 with semantic field type and 38,030 with aggregation. Additionally, we have conducted quality inspection and further checks to ensure accuracy and reliability.

Based on the AnaMeta dataset, in the rest of the paper, we first evaluate a wide range of models on inferring metadata in order to check how well they learned the knowledge of field semantics. In §6.1, pre-trained tabular (TURL, TAPAS, TABBIE) and large language (GPT-3/3.5 family) models are compared through additional classification heads and zero- / few-shot prompts. Semantic information captured by the pre-trained tabular models brings great gain to metadata tasks.

To improve the metadata understanding capability of tabular models, in §4 we further propose a multi-encoder **KDF** (Knowledge and Distribution Fusion) framework for transformer-based pre-trained tabular models. *Knowledge fusion* incorporates knowledge graph information (such as lined entities, column type, and properties). *Distribution fusion* adds distribution information by calculating field data statistics. KDF outperforms the best baseline by 3.13% (see §6.2). This indicates that successful knowledge and distribution fusion brings a better representation of fields.

Finally, we demonstrate one general approach to leveraging metadata to enhance downstream analysis tasks – taking field metadata as an intermediate step and injecting the knowledge through different interfaces for downstream models. In order to incorporate metadata into downstream analysis tasks at various stages of the process, four interfaces are proposed in §5. These interfaces provide different forms of metadata, including field metadata classification results, column embeddings, metadata strings of the field, and field metadata tasks as pretraining objectives. In §6.3, we apply these interfaces on downstream tasks (TableQA and visualization recommendation) and observe that metadata knowledge helps downstream tasks when injected through appropriate interfaces.

In summary, our main contributions are:

- We collect a large AnaMeta dataset with the definition and taxonomy of field metadata and smart supervision. The dataset and code will be open-sourced in `https://github.com/microsoft/AnaMeta`.
- A wide range of models are compared on metadata learning and retaining capabilities.
- We propose a KDF framework with distribution

fusion and knowledge fusion, which bring a better representation of fields.

- Four interfaces are proposed to improve the performance of downstream applications at various stages of the process.

2 Metadata Definition

In this section, we delve deeper into the concept of metadata and its definitions. To facilitate understanding, we provide an example in Figure 1. When creating charts and pivot tables, individuals typically begin by analyzing common fields and assigning different roles to them, *e.g.*, using the primary key – "Product Name" field as the x-axis in a bar chart. After selecting the fields, further analysis is required for each field. In the case of pivot tables, it is necessary to determine that the "original prices" field should be summed. We formulate the metadata tasks as machine learning classification tasks with the table as input. The specifics of this formulation can be found in §B.6.

2.1 Measure / Dimension Dichotomy

Measure and dimension fields play different roles in data analysis. The measure/dimension dichotomy metadata can tell downstream tasks the legal analysis operations for each field, thus could help greatly instruct their search spaces. They are simply defined in (Ding et al., 2019).

Definition 1 (Measure)**.** A **measure** (MSR) field contains numerical measurement values on which calculations can be made.

Definition 2 (Dimension)**.** A **dimension** (DIM) field contains categorical values. It provides functions of filtering, grouping, and labeling. A dimension is called a **breakdown** (or group-by) dimension when its values have duplication. Otherwise, a dimension with unique values is a **key** dimension.

2.2 Common Field Roles

Fields have been identified with measure and dimension, while not all of them are highly regarded. In daily analysis activities, measures and dimensions with some semantic meanings are more frequently selected. In other words, there are common patterns about which fields are more preferred than others within a table. As shown in Figure 1, we mark common preferences of measure, breakdown dimensions, and key dimensions by shade (darker means more preferred).

Definition 3 (Natural Key)**. Natural Key** is a dimension field with all unique data values and uses them to represent each record in semantic terms.

Definition 4 (Common Breakdown)**. Common Breakdown** is the dimension field(s) that are the most commonly used for breaking down (grouping by) among a given table in data analysis.

Definition 5 (Common Measure)**. Common Measure** is the measure field(s) that are the most commonly used for further analysis (*e.g.*, applying aggregation function, composing chart) among a given table in data analysis.

2.3 Semantic Field Type

Semantic types for common fields are important to data cleaning, table interpretation, data analysis, and so on. There are several existing works focusing on identifying semantic types for columns as described in §A.2, *e.g.*, column type identification.

However, most works focus on dimension fields, especially the subject columns. Less than 5% of column types denote measure fields, which are statistics on existing column type datasets (Sherlock (Hulsebos et al., 2019), TURL (Deng et al., 2020), and Semtab (Jiménez-Ruiz et al., 2020; Cutrona et al., 2020)). Thus, we propose semantic field type, including dimension and measure type.

Definition 6 (Dimension Type)**.** The common semantic types for dimension fields, are based on knowledge graph type of entities. We adopt TURL (Deng et al., 2020) column types as dimension type. Details are shown in §B.3

Definition 7 (Measure Type)**.** The common mutually exclusive types for measure fields, are based on unit, magnitudes, and entity property. See the taxonomy in Table 1.

Table 1: Measure Type Taxonomy.

Category	Type	Category	Type
Dimensionless	Count (Amount)	Scientific	Length
	Ratio		Area
	Angle		Volume(Capacity)
	Score		Mass(Weight)
	Rank		Power
	Factor/Coefficient		Energy
Money			Pressure
Data/file size			Speed
Time	Duration		Temperature
	Frequency	Others	

In Table 1 you can find the commonly seen measure types in daily data analysis scenarios. These

19 types (except "Others") can be further grouped into 5 categories. The taxonomy is based on the units in Wikidata[1], the International System of Units[2] and DBPedia properties in T2D Golden Standard[3]. As we will discuss in §3.2, we only keep the measure types with a number of appearances above the threshold. More details in §B.4.

Many tables contain multiple same-typed measures, among which further analysis can be made, implying multi-measure analysis.

Definition 8 (Measure Pair)**.** Within a table, a **measure pair** is a pair of comparable measures – they should have the same type of unit (including convertible ones), related semantic meanings, and a similar numerical value range.

2.4 Default Aggregation Function

Definition 9 (Default Aggregation Function)**. Default aggregation function** is an aggregation function that is the most commonly used for applying to the measure field. Popular aggregation (AGG) functions and their statistics are shown in Figure 4.

3 AnaMeta Dataset

After defining the field metadata, the next challenge is to locate the supervision labels needed for training and evaluation. In this section, we will bring several sources together to prepare the labels.

3.1 Table Sources

3.1.1 Spreadsheet Tables

From the public Web, we crawled millions of Excel spreadsheet files with English-language tables in them. Lots of these files contain analysis artifacts created by users. From these spreadsheets, we extract the following datasets:

1) Chart dataset: There are 59,797 charts (*e.g.*, the one in Figure 1) created from 36,461 tables. Line, bar, scatter, and pie charts are the most dominant chart types, covering more than 98.91% of charts. The x-axes of bar and pie charts directly display the data values of their reference field one by one, which play the role of natural key. Y-axes display data size or trend, which plays the role of measure. And in some charts, the y-axis is plotted with multi-fields, which are measure pair.

2) Pivot dataset: There are 23,728 pivot tables (*e.g.*, the one in Figure 1) created from 8,900 tables. Pivot table has "rows", "columns" and "values". Both the rows and columns hierarchically break down records into groups, which play the role of common breakdown. Its values are for applying aggregation to each group, which plays the role of common measure and provides default aggregation functions.

3.1.2 Web Tables

Several web table (HTML table) datasets have been extracted in prior work, and the following datasets are used in this work:

(1) T2D dataset: T2D Gold Standard (Ritze and Bizer, 2017) is a dataset for evaluating matching systems on the task of matching Web tables to the DBpedia knowledge base. It contains schema-level correspondences between 1,724 Web tables (consisting of 7,705 fields) from the English-language subset of the Web Data Commons Web Tables Corpus (Lehmberg et al., 2016) and DBpedia[4]. In total, more than 2,000 fields are mapped to ∼290 DBPedia properties.

(2) TURL dataset: TURL (Deng et al., 2020) constructs a dataset based on the WikiTable corpus, and utilizes each cell hyperlink (link to Wikipedia pages) to get entity linking, column type, and relation extraction supervision labels. Our work utilizes TURL tables with column types that contain 2,368,990 columns from 403,450 tables.

3.1.3 Synthetic Tables

To utilize richer data and supervision labels, synthetic datasets are also taken into consideration. SemTab challenge (Jiménez-Ruiz et al., 2020) aims at benchmarking systems dealing with the tabular data to knowledge graph matching problem, including the CEA task (matching a cell to a KG entity), the CTA task (assigning a semantic type to a column), and the CPA task (assigning a property to the relationship between two columns). This challenge provides large datasets automatically generated from the knowledge graph. SemTab 2020 dataset contains 131,289 tables and 68,001 tables with CPA task labels. To avoid data leakage, 17,995 schemata are extracted from 68,001 tables, and randomly keep one table in each schema.

[1]https://www.wikidata.org/wiki/Wikidata:Units

[2]https://www.bipm.org/en/measurement-units

[3]http://webdatacommons.org/webtables/goldstandard.html#toc2

[4]https://www.dbpedia.org/

Table 2: Supervision Labels and Statistics from Each Dataset. Each row represents one task, and its labels are obtained from different ways and datasets. The last row ("Statistics") refers to the number of samples with labels.

Supervision	Chart dataset	Pivot dataset	T2D dataset	TURL dataset	SemTab dataset	Statistics (field or pair)
Measure	Y-axis field	Value field	Msr. type field	-	-	110,614
Dimension	X-axis field (ex. scatter, line chart)	Rows and columns field	Primary key field	-	-	41,478
Com. measure	Chart msr.	Pivot msr.	-	-	-	110,352 pos, 299,268 neg
Com. breakdown	-	Non-unique dim.	-	-	-	18,815 pos, 117,996 neg
Natural key	Single unique dim.	-	Primary key	-	-	20,030 pos, 84,496 neg
Dim. type	-	-	-	Column type (cell hyperlinks)	-	1,235,831
Msr. type	-	-	Msr. property (manual label)	-	Msr. property (from synthetic info.)	494,663
Msr. pair	Same chart axis msr.	-	-	-	-	33,100
Agg. function	-	Agg. for value field	-	-	-	38,030
Statistics (schema & table)	21,733 & 36,461	7,041 & 8,900	702 & 702	403,450 & 403,450	17,995 & 17,995	-

3.2 Supervision Labels

Based on the above datasets, supervision labels are prepared in three ways: First, from the analysis artifacts created in downstream tasks including chart and pivot table; Second, from manual labels in the public dataset (*i.e.*, T2D) and by ourselves; Third, from the information attached to the table (*i.e.*, SemTab, TURL). Table 2 shows supervision labels for each task from each dataset.

In Table 2 rows 1-2, for the measure / dimension dichotomy in §2.1, the positive samples (measures) are the fields referenced by y-axis in charts and "values" in pivot tables, the negative samples (dimensions) are x-axis (except scatter and line charts) in charts and "rows" and "columns" in pivot tables.

For the common field roles in §2.2, the labels are slightly modified on measure / dimension dichotomy labels. In Table 2 row 4, for the common breakdown, positive samples come from the non-unique dimensions in pivot tables. In Table 2 row 5, for natural key, samples come from dimensions in charts, and satisfy that 1) data values are unique 2) don't have multi-dimensions in the table. The negative samples are all other fields besides the positive samples in a given table.

In Table 2 row 8, for measure pair in §2.3, its positive labels come from pairs of measure fields referenced by the same chart axis. For each table, we randomly sample the same number of negative samples (numerical field pairs) as positive samples. In Table 2 row 7, for measure type labels, we merge two label sources together: DBPedia property labels in T2D, and Wikidata property labels in SemTab. We have organized the measure type taxonomy and map supervisions from the datasets, and details are shown in §B.4. In row 6, for dimension type labels, we utilize column types in TURL(Deng et al., 2020).

In Table 2 row 9, for default aggregation ranking scores in §2.4, we focus on the 9 most frequently used aggregation functions in pivot tables as shown in Figure 4. For a field, the actual aggregation applied by users has a score = 1, otherwise, unused functions are assigned with 0 score.

3.3 Quality Inspection

To assess the quality of our corpus, we conducted a quality inspection with 5 experts who have analysis experience. We ensured accuracy by informing the annotators of the scoring standards during the annotation process. The score uses ordinal scale values(1, 0.75, 0.5, 0.25, 0) with corresponding definitions. Our results showed that the AnaMeta dataset labels got 0.97 (out of 1) on the macro average with manual annotation, which demonstrates that our corpus contains high-quality data and supervision. Details of inspection and results are shown in §C.2.

4 KDF Framework

As discussed in §1, To improve the metadata understanding capability of tabular models, we propose KDF (Knowledge and Distribution Fusion) framework to infuse information. In this section, we provide a detailed description of Metadata model architecture as depicted in Figure 5.

4.1 Overall Model Architecture

Our model is based on the pre-trained tabular model and utilizes a transformer encoder on top. Because distribution and knowledge graph information describes both cells and columns, we design two encoders for KDF framework. The first encoder is sub-token or cell level (depending on the pre-trained tabular model), and the second one is column level. Details of KDF overall architecture

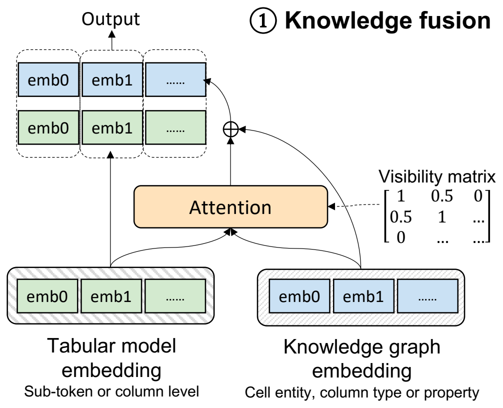


Figure 2: Knowledge Fusion Module in KDF Framework. The module represents each sequence.

are shown in §D.1.

The pre-trained tabular model could be almost any transformer-based pre-trained models. In this work, we use TAPAS(Herzig et al., 2020) and TABBIE(Iida et al., 2021) to illustrate effectiveness.

4.2 Knowledge Fusion Module

Knowledge graph information provides extra knowledge about tables and helps the model interpret tables. In this subsection, we describe how to get that information and how to use them.

Fusion module is illustrated in Figure 2. Model respectively fuses cell entity in cell level, column type, and property type in column level. Because one tabular token needs to pay attention to several entities, *e.g.*, a cell token needs to pay attention to a specific entity in the same column or row, knowledge fusion module applies attention to fuse tabular embedding and knowledge graph embedding, as shown in the following equation.

$$H = W_3\,(\text{Attention}(TOK, ENT, ENT) + ENT)$$
$$\text{Attention}(Q, K, V) = W_2\,\text{softmax}(QW_1K^T + \ln M)\,V$$

where, $TOK \in \mathbb{R}^{n\times d_{tok}}$ is the sequence of tabular model embedding, $ENT \in \mathbb{R}^{n\times d_{ent}}$ is the sequence of knowledge graph embedding, $W_1 \in \mathbb{R}^{d_{tok}\times d_{ent}}$, $W_2 \in \mathbb{R}^{d_{ent}\times d_{ent}}$ and $W_3 \in \mathbb{R}^{d_{ent}\times d_h}$ are trainable weights. $M \in \mathbb{R}^{n\times n}$ is visibility matrix.

The visibility matrix is to control whether an entity is visible to a tabular model token, as shown in Equation (1). Besides, it only works in sub-token (or cell) level fusion.

$$M_{i,j} = \begin{cases} 1 & \text{token } i \text{ and } j \text{ are from the same cell} \\ m & i \text{ is only in the same column or row with } j \\ 0 & \text{others} \end{cases} \tag{1}$$

where, $m \in (0, 1)$ is half visible hyper parameter.

After getting attentive entity embedding (H), we concatenate tabular model embedding (TOK) with it as the module's output.

4.3 Distribution Fusion Module

It is hard for a pre-trained tabular model to capture the entire column distribution, which is important to analyze metadata tasks. To learn numerical distribution, we extract 31 data statistics features and 6 categories for each field (Zhou et al., 2021). In this module, tabular model embedding and field categories are added together after linear layer and embedding lookup respectively. The output of the module is the concatenation of the embedding and statistics features. Details of architecture and features are shown in §D.3.

5 Downstream Interfaces

Field metadata is formulated as classification problems, which can be effectively learned through the KDF framework. However, it remains a challenge to effectively incorporate this metadata into downstream applications. To address this challenge, we propose four interfaces including metadata IDs, embeddings, sentences, and pre-training. These interfaces allow for the integration of metadata at different stages of the analysis process and facilitate its use in downstream applications.

Metadata IDs Interface provides downstream applications with the classification results of field metadata tasks. This interface can be used to provide more specific and targeted uses of field metadata. For example, IDs can be used as rules to limit the scope of downstream searches. During QuickInsights(Ding et al., 2019), different data mining strategies are applied based on whether a field is classified as a measure or dimension. Additionally, ids can be used as tags in language models.

Metadata Embeddings Interface provides downstream applications with each column embeddings, which are augmented with metadata knowledge. These embeddings can represent columns in a continuous, dense, and low-dimensional vector space. Thus they can be used in a variety of data analysis tasks to classify columns, generate analysis according to the column, and so on. Additionally, they can be used as features in downstream machine learning models.

Metadata Sentences Interface provides downstream applications with string sentences, which

Table 3: Benchmarks and KDF Framework Results on Metadata Tasks. All metric numbers are in % and averaged over 3 runs. For each task the **bold** number is the best one. Δ means the best results of "KDF" metric minus the corresponding evaluation metric, and color formatting reacts value size. "TML" means traditional machine learning, "TPLM" means tabular pretrained language model, "LLM" means large language model.

Model		Msr		Natural Key		Com. Breakdown		Com. Measure		Dim Type		Msr Type		Msr Pair		Aggregation	
		Acc.	Δ	HR1	Δ	HR@1	Δ	HR@1	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ
Rule based		96.68	-2.26	88.42	-6.62	49.72	-15.77	67.02	-4.71	12.99	-84.98	25.91	-53.41	32.04	-46.97	90.53	-0.95
TML	GBDT	96.94	-1.99	89.38	-5.66	55.50	-9.99	66.84	-4.90	24.44	-73.53	63.86	-15.47	62.55	-16.46	90.55	-0.94
	RF	98.07	-0.87	92.75	-2.29	59.47	-6.01	67.13	-4.61	45.56	-52.41	66.32	-13.00	63.76	-15.25	90.84	-0.65
TPLM	TURL	97.65	-1.28	92.65	-2.39	62.38	-3.11	70.37	-1.37	96.55	-1.42	66.92	-12.41	69.38	-9.63	91.15	-0.33
	TAPAS	97.09	-1.84	93.79	-1.25	58.62	-6.87	70.01	-1.73	96.21	-1.76	78.49	-0.83	75.20	-3.81	89.96	-1.52
	TABBIE	97.49	-1.45	94.42	-0.62	62.36	-3.13	71.16	-0.58	95.34	-2.63	77.39	-1.94	77.70	-1.32	88.43	-3.05
LLM	GPT-3.5 0-shot	87.18	-11.75	88.39	-6.65	41.51	-23.98	50.18	-21.56	46.11	-51.86	35.10	-44.22	51.45	-27.57	77.46	-14.02
	GPT-3.5 1-shot	91.14	-7.79	82.47	-12.57	44.80	-20.69	57.30	-14.43	46.38	-51.59	29.32	-50.00	55.36	-23.66	75.66	-15.82
KDF	KDF+TAPAS	98.45	-0.48	**95.04**	0.00	64.03	-1.46	71.09	-0.65	**97.97**	0.00	**79.32**	0.00	77.50	-1.52	**91.48**	0.00
	KDF+TABBIE	**98.93**	0.00	94.99	-0.06	**65.49**	0.00	**71.74**	0.00	97.33	-0.64	78.74	-0.58	**79.01**	0.00	90.58	-0.90

describe the metadata knowledge associated with a field. These sentences can provide a human-readable summary of the metadata. As language models with a seq2seq structure have become popular in recent years, using sentences as an interface to combine different models and make use of their knowledge is a current trend. Therefore, by using metadata sentences, it is possible to easily incorporate metadata into various downstream tasks.

Metadata Pretraining Interface provides metadata tasks as pre-training or continue pre-training objectives. This allows for the direct incorporation of metadata knowledge into downstream models during the training stage. Additionally, it should be noted that metadata can be formulated as question-answer tasks to make it more adaptable to a wider range of language models, rather than just classification tasks.

6 Experiments

We conducted three parts of experiments to evaluate field metadata tasks on the AnaMeta dataset and demonstrate how to incorporate analysis knowledge. First, a variety of models are evaluated on all metadata tasks as a benchmark. Second, the KDF framework is evaluated on all metadata tasks Finally, we evaluate the performance of downstream applications using four interfaces. All the experiments are run on Linux machines with 448 GB memory, and 4 NVIDIA Tesla V100 16G-memory GPUs.

6.1 Benchmark

To compare the performance of existing models on all tasks, we use four kinds of baselines – rule-based baseline, traditional machine learning baselines (GBDT and Random Forest), pre-trained tabular model baselines (TURL), and large language model (GPT-3.5: text-davinci-003). Because there is no direct existing experiment on metadata tasks, we slightly change the above models to adapt to our tasks. Details of implementations and GPT-3.5 prompt engineering are shown in §E.3. As described in Table 3, we find the following insights:

Semantic information captured by the pre-trained tabular models brings great gain to metadata tasks. Tabular pre-trained language model outperforms rule-based and traditional machine learning approaches, especially on common field roles and semantic field types (*e.g.* dimension type outperforms about 80% and 60%). Traditional machine learning is good at representing field distribution, while it lacks semantic information and entire table understanding. Pre-trained tabular model fills the gap.

In our preliminary experiments, we investigated the capability of large language models (LLMs) to extract metadata from tables. As shown in Table 3, our results indicate that GPT-3.5 has the ability to extract metadata, though it is not yet ideal. One limitation is that GPT-3.5 may not have been exposed to enough metadata during its training, which can result in performance that is worse than traditional heuristic rules. However, we found that providing GPT-3.5 with one example improved its performance on more than half of the metadata tasks, suggesting that with proper in-context learning, GPT-3.5 could have even better metadata extraction capabilities. Additionally, our initial exploration with naive prompt designs may not have fully leveraged the knowledge present within the large language models. Further research in these areas is warranted.

6.2 KDE Framework

Experiments are conducted on all metadata tasks. We respectively use TAPAS (Herzig et al., 2020) and TABBIE (Iida et al., 2021) as a pre-trained tabular model to illustrate the effectiveness of KDF framework. Details of KDF experiment setting are shown in §E.2. As described in Table 3, we find the following insights across tasks.

Successful distribution and knowledge fusion brings a better representation of fields. KDF models exceed the performance of the other pre-trained tabular models, especially on common field roles, measure type, and measure pair tasks. On the measure type task, KDF (TAPAS) outperforms TURL by 12.41%. Our KDF framework explicitly fuses distribution and knowledge information. Thus, it can better represent the whole field and integrate useful external information. More analysis of the two fusions is described in ablation experiments (§E.6).

On the dimension type task, KDF (TAPAS) achieves the top result and improves over the performance of TURL by 1.42%. It's worth noting that this task is one of TURL's original tasks, while our Metadata model still has advantages. It benefits from better knowledge fusion and pre-training model representation.

6.3 Downstream Interfaces

Table 4: Metadata IDs and Embeddings Interfaces Results on Table2Chart. All metric numbers are averaged over 3 runs.

Interface	Precision	Recall	F1
w/o Metadata	80.60%	77.28%	78.91%
Metadata IDs	81.45%	**77.73%**	79.55%
Metadata Embeddings	**82.44%**	77.56%	**79.92%**

Table 5: Metadata Sentences and Pre-training Interfaces Results on TableQA. All metric numbers are averaged over 3 runs.

Interface	HybridQA	WikiTQ
w/o Metadata	53.69%	36.70%
Metadata Sentences	**54.59%**	**37.83%**
Metadata Pre-training	53.62%	36.14%

In order to evaluate the performance of four interfaces introduced in §5, and demonstrate the importance of field metadata, we conducted experiments using interfaces in several downstream analysis tasks. For the Metadata IDs and embeddings interfaces, which take column features (embeddings) as input, we chose visualization generation tasks and applied the interfaces to the Table2Charts model (Zhou et al., 2021). For the metadata sentences and pre-training interfaces, which are suitable for tasks solved by pre-trained language models with string sentences as inputs, we chose the popular TableQA task and applied the interfaces to the UnifiedSKG framework (Xie et al., 2022). Results can be found in Table 4 and Table 5. More implementation details can be found in §E.7.

Field metadata knowledge can improve downstream analysis tasks when used with the appropriate interface. As seen in Table 4, the Metadata embeddings interface outperforms the baseline with a 1.84% increase in precision. Additionally, in Table 5, the metadata sentences interface improves the TableQA task, even though this task does not directly use metadata as output. These results highlight the importance and necessity of field metadata knowledge learned from field metadata tasks.

Different downstream tasks may benefit from different interfaces depending on factors, such as input, task characteristics, model characteristics, etc. In addition to downstream task inputs that strictly limit the choice of interfaces, the characteristics of the task and model are also important. In Table 5, Metadata sentences interface can boost the task while metadata pre-training gives the model a bad influence. Metadata tasks have different logic of reasoning with TableQA tasks, although it can help understand tables. When using metadata task as pre-training objectives, it destroys the original logic of reasoning and brings side effects.

7 Related Work

7.1 Table Interpretation

There is a long line of work trying to understand tables symbolically, especially for entity or content tables where we can conduct entity linking, column type annotation, and relation extraction (Wang et al., 2012; Kacprzak et al., 2018; Hulsebos et al., 2019; Cutrona et al., 2020; Wang et al., 2021). Some domain ontology or knowledge graph, such as DBPedia (Lehmann et al., 2015) and Wikidata (Vrandečić and Krötzsch, 2014), is often provided for alignment. Column type annotation and relation extraction are related to our measure / dimension (§2.1) and field type (§2.3) classification. However, most previous work focus on the entity type of the columns (Hulsebos et al., 2019;

Deng et al., 2020), and few public datasets provide real-world tables with rich labels of measurements (Ritze and Bizer, 2017; Cutrona et al., 2020).

7.2 Pre-trained Tabular Models

Pre-trained language models (Devlin et al., 2019; Brown et al., 2020) are widely used in NLP tasks. Recently also emerge several pre-trained tabular models with transformer as the primary backbone (Dong et al., 2022), such as, TAPAS (Herzig et al., 2020), TABBIE (Iida et al., 2021) and TURL (Deng et al., 2020). The semantic information within those models contains an amount of metadata knowledge, however, the models still lack the ability to understand things like table distributions, which we demonstrate in §6

8 Conclusion

In conclusion, this paper has presented the AnaMeta dataset, a collection of tables with derived supervision labels for four types of commonly used field metadata. We have also proposed a multi-encoder framework, called KDF, which incorporates distribution and knowledge information. Additionally, we have proposed four interfaces for incorporating field metadata into downstream analysis tasks. Through evaluations of a wide range of models, we have shown the importance of accurate metadata understanding for tabular data analysis and the effectiveness of the KDF framework and interfaces in improving the performance of downstream tasks.

Limitations

The type of field metadata tasks is limited in this paper and it can be explored more. There are far more types of analysis metadata to be discovered and inferred. On the one hand, inspired by data profiling metadata, the dependency between multi-fields in one table plays an important role. There are several common dependencies or relationships among columns. How to identify them is future work. On the other hand, in Table 8 only a limited taxonomy is provided. A more comprehensive one is future work.

Our initial research explored the ability of large language models (LLMs) to extract metadata from tables. The results were not optimal, likely due to a lack of exposure to metadata during the training process of the LLM and limitations in the design of the prompts used. Further investigation is necessary to improve the performance of LLMs in extracting metadata from tables.

Ethical Statements

We collect AnaMeta dataset from 3 source datasets. For spreadsheet dataset, we crawl them from websites, which means original spreadsheet are public. Thus, We believe there is no privacy issue related to this dataset. For web tables and synthetic tables, we apply public datasets and follow their License.

References

Iso 80000-1:2009 quantities and units. https://www.iso.org/standard/30669.html. Accessed: 2021-09-15.

Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. 2018. Data profiling. *Synthesis Lectures on Data Management*, 10(4):1–154.

Mesures BIdPe. 2019. Si brochure: The international system of units (si). *ed. https://www.bipm.org 2019*.

Leon Bornemann, Tobias Bleifuß, Dmitri V Kalashnikov, Felix Naumann, and Divesh Srivastava. 2020. Natural key discovery in wikipedia tables. In *Proceedings of The Web Conference 2020*, pages 2789–2795.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Vincenzo Cutrona, Federico Bianchi, Ernesto Jiménez-Ruiz, and Matteo Palmonari. 2020. Tough tables: Carefully evaluating entity linking for tabular data. In *The Semantic Web – ISWC 2020*, pages 328–343. Springer International Publishing.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. Turl: Table understanding through representation learning. *Proc. VLDB Endow.*, 14(3):307–319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Rui Ding, Shi Han, Yong Xu, Haidong Zhang, and Dongmei Zhang. 2019. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 317–332. Association for Computing Machinery.

Haoyu Dong, Zhoujun Cheng, Xinyi He, Mengyu Zhou, Anda Zhou, Fan Zhou, Ao Liu, Shi Han, and Dongmei Zhang. 2022. Table pre-training: A survey on model architectures, pre-training objectives, and downstream tasks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 5426–5435. Survey Track.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43. Association for Computational Linguistics.

Matteo Golfarelli, Dario Maio, and Stefano Rizzi. 1998. The dimensional fact model: A conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, 7(02n03):215–247.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333.

Jamie Hoelscher and Amanda Mortimer. 2018. Using tableau to visualize data and drive decision-making. *Journal of Accounting Education*, 44:49–59.

Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, KDD '19, page 1500–1508. Association for Computing Machinery.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456.

Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, and Kavitha Srinivas. 2020. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In *European Semantic Web Conference*, pages 514–530. Springer.

Emilia Kacprzak, José M. Giménez-García, Alessandro Piscopo, Laura Koesten, Luis-Daniel Ibáñez, Jeni Tennison, and Elena Simperl. 2018. Making sense of numerical data - semantic labelling of web tables. In *Knowledge Engineering and Knowledge Management*, pages 163–178. Springer International Publishing.

George Katsogiannis-Meimarakis and Georgia Koutrika. 2021. *A Deep Dive into Deep Learning Approaches for Text-to-SQL Systems*, page 2846–2851. Association for Computing Machinery.

R. Kimball and M. Ross. 2013. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley.

Po-Ming Law, Alex Endert, and John Stasko. 2020. What are data insights to professional visualization users? In *2020 IEEE Visualization Conference (VIS)*, pages 181–185.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 75–76, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

P. Nguyen, N. Kertkeidkachorn, R. Ichise, and Hideaki Takeda. 2019. Mtab: Matching tabular data to knowledge graph using probability models. In *SemTab@ISWC*.

Carl M Rebman Jr, Queen E Booker, Hayden Wimmer, Steve Levkoff, Mark McMurtrey, and Loreen Marie Powell. 2022. An industry survey of analytics spreadsheet tools adoption: Microsoft excel vs google sheets. In *Proceedings of the EDSIG Conference ISSN*, volume 2473, page 4901.

Dominique Ritze and Christian Bizer. 2017. Matching web tables to dbpedia - a feature utility study. In *EDBT*, pages 210–221. OpenProceedings.org.

Sunita Sarawagi and Soumen Chakrabarti. 2014. Open-domain quantity queries on web tables: annotation, response, and consensus models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 711–720.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010.

Tobias Vogel and Felix Naumann. 2011. Instance-based ‘one-to-some’assignment of similarity measures to attributes. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 412–420. Springer.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. 2021. Tcn: Table convolutional network for web table interpretation. *Proceedings of the Web Conference 2021*.

Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q. Zhu. 2012. Understanding tables on the web. In *Conceptual Modeling*, pages 141–155. Springer Berlin Heidelberg.

Aoyu Wu, Yun Wang, Xinhuan Shu, Dominik Moritz, Weiwei Cui, Haidong Zhang, Dongmei Zhang, and Huamin Qu. 2021. Ai4vis: Survey on artificial intelligence approaches for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *EMNLP*.

Yang Yi, Zhiyu Chen, Jeff Heflin, and Brian D Davison. 2018. Recognizing quantity names for tabular data. In *ProfS/KG4IR/Data: Search@ SIGIR*.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing.

Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M Procopiuc, and Divesh Srivastava. 2011. Automatic discovery of attributes in relational databases. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 109–120.

Shuo Zhang and Krisztian Balog. 2017. Entitables: Smart assistance for entity-focused tables. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’17, page 255–264. Association for Computing Machinery.

Shuo Zhang and Krisztian Balog. 2019. Auto-completion for data cells in relational tables. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, page 761–770. Association for Computing Machinery.

Shuo Zhang and Krisztian Balog. 2020. Web table extraction, retrieval, and augmentation: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(2).

Mengyu Zhou, Qingtao Li, Xinyi He, Yuejiang Li, Yibo Liu, Wei Ji, Shi Han, Yining Chen, Daxin Jiang, and Dongmei Zhang. 2021. Table2charts: Recommending charts by learning shared table representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD ’21, page 2389–2399.

Mengyu Zhou, Wang Tao, Ji Pengxin, Han Shi, and Zhang Dongmei. 2020. Table2analysis: Modeling and recommendation of common analysis patterns for multi-dimensional data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):320–328.

A Related Work

A.1 Dimensional Modeling and Metrology

The terms "measure" and "dimension" have their roots in dimensional modeling from data warehousing and business intelligence (Golfarelli et al., 1998). In relational and multidimensional databases, dimensional models are implemented as star schemas and online analytical processing (OLAP) cubes (Kimball and Ross, 2013). Our definition of "dimension" extends the concept in dimensional modeling. As we will discuss in §2.1, it contains primary keys and natural keys in addition to the dimension attributes. Most of our analysis metadata involves measures. As the scientific study of measurements, Metrology includes the definition of quantities and units of measurement. In §2.3, we will define our measure types with common units from the International System of Units (SI) (BIdPe, 2019; iso), which is a widely accepted metric system.

A.2 Data Profiling

(Abedjan et al., 2018) has the similar term "metadata". These metadata, such as statistics about the data or dependencies among columns, can help understand and manage new datasets.

Analysis metadata is proposed for further analysis tasks based on part of data profiling metadata. The details about their relationship for each task are as follows: (1) Unique column combination and primary key. There is plenty of work to explore them, while key with semantics is not their focus. (Bornemann et al., 2020) proposed natural key based on primary key, and they use engineered features and Random Forest to solve the problem.

(2) Identifying semantic domain of a column. (Zhang et al., 2011) first propose semantic domain labeling by clustering columns. (Vogel and Naumann, 2011) matches columns to pre-defined semantics with specific features and Naive Bayes. This track of works evolves towards column type identification in table interpretation, which we discuss in §7.1.

(3) Quantity name recognition: This activates detecting the quantity name for a column, which is highly correlated with measure type in analysis metadata. (Sarawagi and Chakrabarti, 2014) point out that unit extraction is a significant step for queries on web tables, and design unit extractors for units in column names by developing a unit catalog tree. (Yi et al., 2018) extends to inferring unknown units with extracted feature and Random Forest. However, those existing works heavily depend on units appearing in the table, so we propose measure types that can also identify measure fields without units and property.

Table 6: Numerical Dimension.

Header	Records
Class	1,2,3,4,5...
Rank	11,12,13,14,15...
ID	9131115,22112723,1111145,30320912...
QP Code	1256,1245,1237,2134...
Style	4,4,2,2,2...

State-Of-Art of those related works often extracts specific features and uses traditional machine learning to solve the problem, which lacks further semantic representation with the pre-training model.

A.3 Downstream Analysis Tasks

Lots of intelligent data analysis features could benefit from analysis metadata. Typical examples include automatic insights discovery (Ding et al., 2019; Law et al., 2020), chart and pivot table recommendations (Zhou et al., 2020, 2021; Wu et al., 2021), Text2SQL and query recommendations (Dong and Lapata, 2016; Katsogiannis-Meimarakis and Koutrika, 2021; Yu et al., 2021), table expansion (Zhang and Balog, 2017, 2019), *etc.* Most of these tasks involve searching, enumerating, and comparing in a large space. Analysis metadata could help narrow down possible candidates (prioritized searching order) and provide good ranking references.

B Problem Definition

B.1 Measure / Dimension Dichotomy

First, not all numerical fields are measure fields. In Table 6, "Style" fields consist of categorical numbers, thus is dimension. "ID" and "QP Code" fields represent keys, thus are dimensions. Second, there exists a weak dependency between field positions and roles. Starting from the left, usually, key dimensions come before breaking down dimensions, and measures come after dimensions. An ML model should take vague hints among fields into account.

B.2 Common Field Roles

It's worth noting that there are several existing works on "primary key", while "natural key" is different from "primary key" as described in §A.2. Both of them are chosen to represent records, while "primary key" focuses on unique (*e.g.*, ID) and "natural key" focuses on semantic terms (*e.g.*, name). Sometimes it is also called key / core / subject / name / entity column in relational tables (Ritze and Bizer, 2017; Zhang and Balog, 2020). As discussed in §7.1, natural key can be used for entity linking and other table understanding tasks.

B.3 Dimension Type

We adopt TURL (Deng et al., 2020) column types as dimension type. It contains 255 types and the most frequency dimension types are shown in Table 7.

Table 7: The Most Frequency Dimension Types.

Dimension type	
people.person	government.political_party
location.location	location.administrative_division
organization.organization	sports.sports_league_season
sports.sports_team	soccer.football_player
sports.pro_athlete	sports.sports_league
soccer.football_team	government.politician
time.event	film.film
location.country	business.business_operation
location.citytown	...

B.4 Measure Type

Each type represents a magnitude, and each type is mutually exclusive. Each measure type corresponds to a set of convertible units (see "Common Units" in Table 8), and highly correlated concepts (see "Common Examples" in Table 8). "Dimensionless" category with 6 types is summarized in the taxonomy. Different from the existing magnitude or units taxonomy, there is plenty of measure without units and corresponding measure types are important in analysis. Thus we summarize the most common mutually exclusive 6 dimensionless measure types.

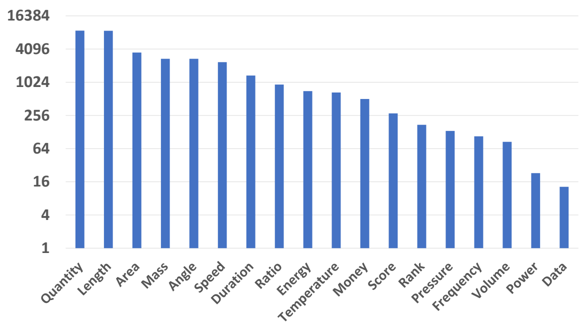


Figure 3: Measure Type. (Taking the logarithm of y-axis with base 4)

The field still has a measure type, when values of the measure field are not entity or property in KG. Measure type is important to all common measure fields. But the existing column type identification and relation extraction task need table values to be entities or properties in the knowledge graph. And more than 90% of numerical fields can not match entity or property in KG (we perform statistics on non-WikiTable (Web table and spreadsheet) mentioned in §3 and match with MTab(Nguyen et al., 2019)). For example, "Final Exam" in Table **??** can not match KG, while it has "Score" measure type.

When determining the measure type taxonomy, we also collect manual labels on 882 tables (consisting of 6,715 fields) randomly sampled from our spreadsheet dataset. We start with a longer list of measure types as discussed in Definition7, map DBPedia properties to the list for T2D, map Wikidata properties for SemTab and mark all 3,139 measures with types in the 882 sampled tables. As mentioned in §2.3, we only keep measure types with ≥10 labels (in T2D and sampled tables) or ≥100, resulting in 36,859 fields fall in our measure taxonomy in Table 8. Although TURL dataset also has property (relation extraction in its paper) labels, there are less than 1% labeled as measure property, so they are not used in this task.

B.5 Default Aggregation Function

An essential operation in data analysis is to aggregate multiple measurements from a field, usually grouped by another breakdown dimension. For each measure, there are some AGG functions more widely applied to it. The most suitable AGG function could be adopted as default calculation by downstream analysis tasks. For most measures, AVG can be applied directly, but often SUM is a better choice if possible. For some downstream scenarios, AVG/SUM can cover most usage cases.

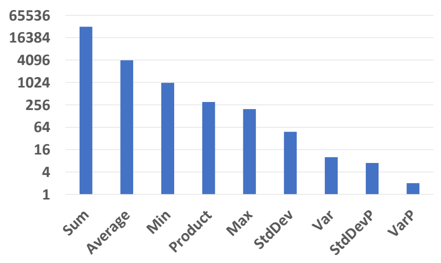


Figure 4: Default Aggregation Functions. (Taking the logarithm of y-axis with base 4)

B.6 Problem Formulation

The machine learning formulation of the metadata is a Binary classification of measure/dimension for

Table 8: Measure Type Taxonomy and Details.

Category	Type	Common Examples	Common Units
Dimensionless	Count (Amount)	Population, daily passenger flow, ...	
	Ratio	Percentage, change rate, proportion, ...	
	Angle	Angle, longitude, latitude ...	
	Factor / Coefficient	Coefficient of thermal expansion, drag coefficient, ...	
	Score	Rating, exam score, indicator (index), ...	
	Rank	University ranking, projected GDP ranking, ...	
Money		Sales, asset, income, revenue, cost, GDP, ...	\$, €, £, ...
Data/file size		Memory size, disk size, ...	GB, kb, ...
Time	Duration	Age, runtime, time length, ...	s, min, hr, d, yr, ...
	Frequency	Audio frequency, rotational speed, ...	Hz, RPM, ...
Scientific	Length	Length, width, elevation, depth, height, ...	m, cm, yard, feet, ...
	Area	Surface area, gross floor area, ...	m^2, acre, ...
	Volume (Capacity)	Vital capacity, water capacity, ...	m^3, L, ...
	Mass (Weight)	Body weight, salt consumption, ...	kg, lbs, ...
	Power	Source power, rated power, ...	kW, ...
	Energy	Calories, energy consumption, ...	J, kcal, ...
	Pressure	Atmospheric pressure, blood pressure, ...	Pa, mmHg, ...
	Speed	Velocity, average speed, ...	m/s, km/h, ...
	Temperature	Effective temperature, melting point, boiling point, ...	$^\circ C$, K, ...

each field of a given table.

To help downstream tasks on prioritizing search order and re-ranking results, we formulate common field roles as three machine learning tasks providing 0$\sim$1 commonness score (higher means more preferred in general) for each field of a given table and recommend fields list for each table in order of commonness score.

The machine learning task for measure pair identification is a binary classification of any pair of numerical fields within a given table. For measure type and dimension type, it is a 19-way and 255-way classification problem.

The machine learning task for the default aggregation function is to provide 0$\sim$1 ranking scores for popular AGG functions.

C Corpus

C.1 Dataset Preprocessing

To avoid data leakage and imbalance, we carry out the following steps. Note that for a fair comparison, we adopt TURL dataset train/valid/test split, so do not perform the following steps.

1. *Table Deduplication.* To avoid the "data leakage" problem that duplicated tables are allocated into both training and testing sets, tables are grouped according to their schemas[5].
2. *Down Sampling.* After deduplication, the number of tables within each schema is very imbalanced – 0.23% schemas cover 20% of these tables. To mitigate this problem, we randomly sample unique tables under the threshold (11 for the Chart dataset, 2 for the Pivot dataset, 1 for Vendor & T2D & Semtab dataset).

[5]Two tables are defined to have the same **schema** if they have the same number of fields, and each field's data type and header name are correspondingly equal.

The schemas are randomly allocated for training, validation, and testing in the ratio of 7:1:2.

For spreadsheet datasets, we follow the steps in (Zhou et al., 2021) and (Zhou et al., 2020), including extraction charts and pivot tables.

For all six datasets, we extract data features (§4.3) and map knowledge graph (§4.2) as the input of our models.

C.2 Data Quality

The inspection focused on the Chart and Pivot datasets, as the quality of the T2D, TURL, and SemTab datasets had already been checked by their authors (for tasks with different labels, we also performed a manual mapping). We conducted the quality inspection with 5 experts who have analysis experience. All experts are from China. During the inspection, we randomly selected 100 tables from the Chart and Pivot datasets, and asked experts to score the labels of the corresponding tasks, and the score is between 0 and 1. Our results showed that measure / dimension dichotomy got 0.99, common field roles got 0.97, aggregation functions got 0.93 and measure pair got 0.97 on average. This demonstrates that our corpus contains high-quality data and supervision.

D KDF Framework

D.1 Overall Model Architecture

KDF overall architecture is as follows:

(1) Pre-trained tabular models are used as preliminary encoders generating initial representation for table elements. After the preliminary encoding phase(**"Pre-trained tabular model"** in Figure 5), we get a sub-token level or cell level (according to different pre-trained tabular models) table embedding sequence.

(2) We fuse knowledge of cell entity and tabular model embedding with **"Knowledge fusion"**, and pass it into sub-token (or cell) level Transformer(Vaswani et al., 2017) encoder(**"Sub-token level encoder"**).

(3) We apply **average pooling** to get an embedding representation for each column. For each column, we use **"Distribution fusion"** to fuse distribution from data features and **"Knowledge fusion"** to fuse knowledge of column type and property in order. Then we pass those column embeddings into Transformer encoder (**"Column level encoder"**) and linear output heads for each metadata task.

D.2 Knowledge Fusion Module

For both entity and property, existing knowledge representation covers comprehensively and performs well. However, if their embedding is trained with table corpus, it can only cover limited entity and property. To increase the extensibility and performance of the model, we directly use knowledge representation in OpenKE[6].

There are three kinds of knowledge information linked with knowledge graph according to table interpretation tasks – cell entity, column type, and property. For a table with knowledge linking (*e.g.* TURL dataset, Semtab dataset), we utilize original knowledge linking. Otherwise, we adopt MTab[7] to link the knowledge graph. MTab is a tool to annotate tables with knowledge graphs, and they get first place in Semtab2019(Jiménez-Ruiz et al., 2020) and Semtab2020(Cutrona et al., 2020).

D.3 Distribution Fusion Module

(1) Statistics features:

- Progression features: ChangeRate, PartialOrdered, OrderedConfidence, ArithmeticProgressionConfidence, GeometricProgressionConfidence.
- String features: AggrPercentFormatted, medianLen, LengthStdDev, AvgLogLength, CommonPrefix, CommonSuffix, Cardinality, AbsoluteCardinality.
- Number range features: Aggr01Ranged, Aggr0100Ranged, AggrInteger, AggrNegative, SumIn01, SumIn0100.
- Distribution features: Benford, Range, NumRows, KeyEntropy, CharEntropy, Variance, Cov, Spread, Major, Skewness, Kurtosis, Gini.

(2) Field categories:

Including FieldType (Unknown, String, Year, DateTime, Decimal), IsPercent, IsCurrency, HasYear, HasMonth and HasDay.

D.4 Multi-task Learning

We use 19-class Cross Entropy loss for measure type, and standard binary Cross Entropy loss for measure/dimension dichotomy, common measures and dimensions, and measure pair tasks. In common measures and dimensions task, we use scores of "true" class as their commonness scores. For aggregation function and dimension type task, we respectively use 7- and 255-class Cross Entropy loss and average loss of each field by the number of its ground truth, because there may be more than one ground truth for one field, which will lead to an unbalanced loss without averaging the fields.

E Experiment Details

E.1 Experiment Setting

All the experiments are run on Linux machines with 24 CPUs, 448 GB memory, and 4 NVIDIA Tesla V100 16G-memory GPUs. We use hit rate ($HR@k = \frac{\#hits@k}{\#samples}$) to evaluate recommendation tasks on each table and accuracy to evaluate classification tasks on each field.

It's worth noting that to avoid label leaking, we do not use Knowledge fusion for column type and property in dimension type task. For the measure type task, we focus on the columns that can not be directly linked with properties from the knowledge graph for these columns are hard for existing table interpretation methods. To imitate those tables, we mask knowledge graph information for the field to train and evaluate in measure type task.

E.2 Metadata Model Details

In detail, we have the following model size:

- Sub-token level: transformer encoder: 2 layers, 8 heads, dimension of embedding: d_{tok} = 192,

[6]http://openke.thunlp.org/

[7]https://github.com/phucty/mtab_tool

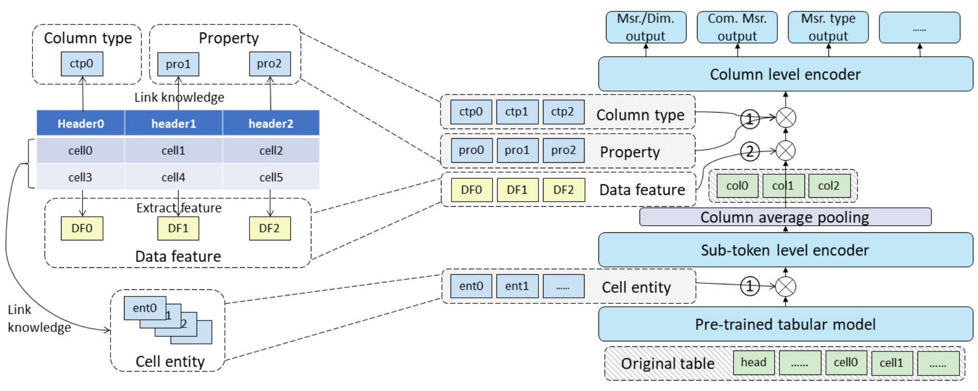


Figure 5: Overall metadata model. ① represents knowledge fusion module in Figure 2, and ② represents distribution fusion module in Figure 6.

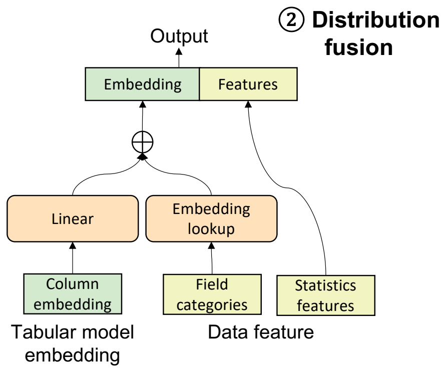


Figure 6: Distribution Fusion Module in KDF Framework. The module represents each token.

d_{ent} = 100, d_h = 64

- Column level: transformer encoder: 2 layers, 8 heads, dimension of embedding: d_{tok} = 128, d_{ent} = 100, d_h = 64

We train Metadata on 10 epochs, 64 batch sizes, AdamW optimizer with 1×10^{-4} learning rate and 0.001 weight decay. We choose half visible hyperparameter $m = 0.5$. It's worth noting that, due to insufficient supervision, aggregation function tasks use the result of epoch5.

E.3 Benchmarks

E.3.1 Rule-based

The rule-based model predicts an output for every field based on the field surface attributes since it is off-the-shelf ready to use and does not involve additional training. The specific rules for each metadata task are as follows:

(1) Measure/dimension dichotomy: The model predicts Measure when the input field is numerical (*i.e.*, the field consists purely of numbers); otherwise; the model predicts Dimension.

(2) Nature key: The model predicts a field to be Natural Key if and only if the field is the leftmost field among fields with cardinality of 1 (*i.e.*, the field contains all unique data values).

(3) Common breakdown: The model predicts a field to be Common Breakdown if and only if the field is the leftmost among dimension fields whose cardinality is less than 0.4.

(4) Common measures: The model predicts a field to be Common Measure if and only if the field is the rightmost numerical field.

(5) Dimension type: The model predicts every field to be of the type with most samples (sports.sports_team in our case).

(6) Measure type: The model predicts every field to be of the type with the most samples (Count in our case).

(7) Measure pair: The model predicts two fields to be a Measure Pair if and only if they are contiguous Measure fields.

(8) Default Aggregation Function: The model predicts the default aggregation function of every field to be the function with most samples (SUM in our case).

E.3.2 Traditional Machine Learning

Traditional machine learning is widely used due to both its effectiveness and efficiency. In data profiling(Abedjan et al., 2018), there exist several state-of-art works that apply traditional learning with specifically extracted features. To compare with them, we design a traditional machine learning baseline by adopting a series of manually designed field data features (elaborated in §4.3). Traditional machine learning baseline implementation details of each metadata task are as follows:

(1) In binary classification tasks (*i.e.*, measure/dimension dichotomy, measure pair), a single classification model is trained, and the model hyperpa-

rameters are determined based on the validation set metrics.

(2) In ranking tasks(*i.e.*, common measures, common breakdown, and primary key), we train a binary classifier to make predictions for each field and rank the predictions with the raw probabilities given by the model outputs.

(3) In multi-class classification tasks (*i.e.* dimension type, measure type, and default aggregation function), we directly train a multi-class classifier using traditional machine learning algorithms.

In the experiment, we choose GBDT, Random Forest, Adaboost, and Naive Bayes to evaluation on each task, and display the performance of the best two baselines (GBDT and Random Forest) to compare with Metadata model.

E.3.3 Pre-trained Tabular Model

TURL is a structure-aware Transformer encoder to model the structural information about tables on table interpretation tasks and achieves Sate-Of-Art results. They also use Masked Entity Recovery pre-training objective to learn knowledge information about entities in relation tables. Since this model considers entity information, we select it as a strong baseline to show the effectiveness of Metadata model architecture in fusing the knowledge graph information.

To evaluate the performance of TURL in metadata tasks and have a fair comparison, we take the following steps:

(1) Map knowledge entity. Because entity linking in Semtab2020 dataset only has Wikidata id but TURL uses DBPedia id in embedding, we use WikiMapper[8] to map Wikidata id to DBPedia id. It's worth noting that dimension type task is one of TURL's original tasks, so WikiMapper is not used in this task. Thus, this task can prove that Metadata still exceeds TURL without the influence of WikiMapper (details are shown in §6.2).

(2) Interpret the original table. We use TURL pre-trained encoder[9] to interpret the original table and get the embedding representation of flat tables.

(3) Train and evaluate metadata tasks. To do classification on metadata tasks, we use the same sub-token level encoder, column level encoder and loss function as our Metadata model while adopting the same training strategy with TURL.

For TAPAS and TABBIE, we use the same sub-token level encoder, column level encoder in Table 5 (without fusion modules).

[8]https://github.com/jcklie/wikimapper

[9]https://github.com/sunlab-osu/TURL

E.3.4 Large Language Model

LLMs have recently gained significant attention for their success in various downstream tasks. In this study, we conduct a preliminary exploration of LLMs performance on metadata tasks.

Hyperparameter Setting We set the decoding temperature to 0 (i.e., greedy decoding without sampling) since the question of metadata tasks has a unique answer. We set the max output length to 200.

Models We select GPT-3 (text-davinci-001) and GPT-3.5 (text-davinci-002 and text-davinci-003) to conduct experiments since these models are shown to be powerful in many downstream tasks.

Zero / Few-shot Learning We apply zero-shot and few-shot learning to evaluate the performance of large language models on metadata tasks. For each setting, we add the definition of metadata terminology to allow LLMs for in-context learning. Besides, the few-shot example is uniformly sampled from the training set for each task. Only high-quality examples with a non-empty header are selected.

Prompts We have created specific prompts for various tasks, which are listed at the end of this section. The optional '<example></example>' block includes a few-shot example. Additionally, for the dimension type task with a large number of hierarchical choices, we employ a hierarchical questioning approach to avoid truncated input. For instance, we furnish the first-layer options (e.g. people, location, organization...) to the LLMs and the second-layer options (e.g. sports_team, pro_athlete, sports_league...) to those questions answered with the correct first-layer choices.

```
**Task: measure / dimension**
<example>
Given the markdown table:
[linearized markdown table row by row]
Is the [ordinal number] column ("[header
 name]" column) a measure or dimension?
The answer is: [answer]
</example> (optional)
Given the markdown table:
[linearized markdown table row by row]
Is the [ordinal number] column ("[header
 name]" column) measure or dimension? ([
term definition]) Please answer
concisely a 'measure' or 'dimension'. (
Do not return any explanation or any
additional information.)
=>
```

```
**Task: natural key, common breakdown,
common measure**
<example>
Given the markdown table:
[linearized markdown table row by row]
Which column is the natural key / common
 breakdown / common measure? The answer
is: [answer]
</example> (optional)
Given the markdown table:
[linearized markdown table row by row]
Which column is the natural key / common
 breakdown / common measure with the
highest probability? ([term definition])
 Please answer a tuple of '(ordinal
English word, header name)', where '
ordinal English word' starts from 'first
'. (Do not return any explanation or any
 additional information.)
=>

**Task: dimension type, measure type,
aggregation**
<example>
Given the markdown table:
[linearized markdown table row by row]
Which dimension type / measure type /
aggregation is the [ordinal number]
column ("[header name]" column)? The
answer is: [answer]
</example> (optional)
Given the markdown table:
[linearized markdown table row by row]
Which of the following dimension types /
 measure types / aggregations is the [
ordinal number] column ("[header name]"
column)? (Do not return any explanation
or any additional information.)
[choice 1]
[choice 2]
...
=>

**Task: measure pair**
<example>
Given the markdown table:
[linearized markdown table row by row]
Is the [ordinal number] column ("[header
 name]" column) and the [ordinal number]
 column ("[header name]" column) a
measure pair? The answer is: [answer]
</example> (optional)
Given the markdown table:
[linearized markdown table row by row]
Is the [ordinal number] column ("[header
 name]" column) and the [ordinal number]
 column ("[header name]" column) a
measure pair? [term definition] Please
answer concisely a 'yes' or 'no'. (Do
not return any explanation or any
additional information.)
=>
```

E.4 Results of Large Language Models

In Table 9, there are more results on large language models for metadata tasks. We only report the results without terminology definition since we observe no significant improvement but degraded performance on measure / dimension dichotomy (-1% on 0-shot, -5% on 1-shot) and natural key (-4% on 0-shot, -1% on 1-shot) tasks on text-davinci-003.

E.5 Measure Type

As mentioned in §A.2, there are several existing works focusing on the similar or same task. To compare dimension type, we apply the State-Of-Art table interpretation model – TURL as a baseline as discussed in §E.3.3. To compare measure type, we apply the most relevant unit detection work (Yi et al., 2018) (RQN) as a baseline. It proposes a feature-based method to automatically determine the quantity names for column values. It uses hand-designed rules to extract features from raw tables in both column values and name. The extracted features are used to train a random forest classifier. The results are shown in Table 10.

In Table 10, Metadata models outperform RQN more than 10%. RQN has a limit to identifying quantities from the existing unit in the column, while Metadata model learns more.

E.6 Ablation Results

We ablate Distribution Fusion and Knowledge Fusion respectively and together, and the results are shown in Table 11.

Successful distribution and knowledge fusion gain the performance of models together. Compared with ablation models, for both TAPAS and TABBIE, the top-1 scores come most frequently from the non-ablation Metadata model. Especially, KG and DF boost 5.40 % (3.13 %) for common breakdown task with Metadata TAPAS (TABBIE), 1.76% (1.99%) for dimension type, and 2.29% (1.32%) for measure pair. For example, on the measure pair task, all measure fields are numerical fields, so number understanding is the key point for this task. Data features fuse statistics information to help the model understand the full picture of numbers in one column. knowledge graph provides existing knowledge to understand each record and column and helps the model to understand the entire table, which is important for measure pair task.

Especially on Common Goup By and Common Measure tasks, data statistics feature information is of great help to understand tables. For Common breakdown tasks, DF improves performance by 2.36% (2.85%) on Metadata TAPAS (TABBIE).

Table 9: Large Language Model More Results on Metadata. “text-davinci-001” is GPT-3, and “text-davinci-002” and “text-davinci-003” are GPT-3.5.

Model	Msr	Natural Key	Com. Breakdown	Com. Msr	Dim Type	Msr Type	Msr Pair	Aggregation
	Acc.	HR@1	HR@1	HR@1	Acc.	Acc.	Acc.	Acc.
text-davinci-001 0-shot	28.28%	73.66%	38.79%	26.03%	4.43%	13.10%	55.71%	19.49%
text-davinci-001 1-shot	52.86%	78.01%	38.48%	22.85%	2.59%	10.92%	54.11%	35.84%
text-davinci-002 0-shot	64.57%	88.10%	44.23%	43.08%	31.85%	18.64%	58.41%	65.26%
text-davinci-002 1-shot	79.41%	74.03%	40.81%	57.36%	35.94%	16.37%	55.63%	55.01%
text-davinci-003 0-shot	87.18%	88.39%	41.51%	50.18%	46.11%	35.10%	51.45%	77.46%
text-davinci-003 1-shot	91.14%	82.47%	44.80%	57.30%	46.38%	29.32%	55.36%	75.66%
text-davinci-003 3-shot	90.53%	78.81%	45.12%	54.14%	41.12%	28.43%	56.81%	80.37%
text-davinci-003 5-shot	90.71%	79.23%	44.82%	56.44%	40.48%	29.18%	56.96%	80.47%

Table 10: Measure Type Additional Results.

Model	Measure Type Acc.
RQN	65.55%
RF	70.09%
KDF+TAPAS w/o DF	**81.51%**
KDF+TABBIE w/o DF	80.29%

Breakdown often appears in long tables (*e.g.* average row number in the pivot dataset is 5805), due to the limitation of sequence length and comprehension, it’s hard to capture the distribution of values for the pre-trained tabular model. Statistics feature is a good choice for understanding distribution. It is worth noting that DF does not benefit all tasks, *e.g.*, measure type task.

Especially on dimension type and measure type tasks, knowledge graph information is of great help to understand tables. For measure type task, KG improves performance by 3.02% (2.90%) on Metadata TAPAS (TABBIE) (w/o DF - w/o DF KG). The model could get additional information from the knowledge graph to enhance the understanding of the table. And it learns the useful pattern between measure type and cell entity/column type.

The replaceable pre-trained tabular model brings a good opportunity to get better performance. KDF(TABBIE) outperforms KDF(TAPAS) on most tasks. Besides, KDF framework outperforms TURL on all tasks. In addition to distribution and knowledge infusing, the replaceable pre-trained tabular model is another key point. More and more pre-trained tabular models are emerging with better performance, and KDF model can be based on almost all transformer-based pre-training models. Thus, it’s convenient to replace them with the models in Metadata, which can further improve performance and choose the best one.

E.7 Downstream Interfaces

In our experiments on the Table2Charts model, we followed the same setup as described in (Zhou et al., 2021) and reported the results of the pretraining stage. In the metadata embeddings interface experiments, we used column embeddings from the KDF framework instead of FastText embeddings in the Table2Charts model. To ensure fair comparisons, we used TAPAS column embeddings as the input for the Table2Charts model, as the only difference between TAPAS column embeddings and KDF embeddings is the use of metadata knowledge. In the metadata IDs interface experiments, we used metadata ID tags on TAPAS column embeddings as the input for a fair comparison with the above experiments.

For TableQA tasks, we followed the T5-base experiment setup of UnifiedSKG (Xie et al., 2022). In the metadata sentences interface experiments, we first used our best KDF model to infer WikiTQ and HybridQA tables. Then, we concatenated metadata sentences, such as [Measure], [Natural Key], [Sum], etc., after the column headers. Finally, we fine-tuned the T5-base model using the same approach as in UnifiedSKG. In the metadata pretraining interface, we first continued pretraining the T5-base model using metadata tasks formulated as table question-answer tasks. The questions and answers were the same as the prompts described in §E.3.4. Then, we fine-tuned the T5-base model using the same approach as in UnifiedSKG. We experimented with all combinations of four metadata tasks and reported the best results.

Table 11: Ablation Results. Setting of this table is the same as Table 3. "DF" – Distribution fusion, "KG" – Knowledge fusion. Δ means non-ablation models (KDF+TAPAS or KDF+TABBIE) metric minus corresponding evaluation metric.

Model	Msr		Natural Key		Com. Breakdown		Com. Measure		Dim Type		Msr Type		Msr Pair		Aggregation	
	Acc.	Δ	HR@1	Δ	HR@1	Δ	HR@1	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ	Acc.	Δ
KDF+TAPAS	98.45	0.00	**95.04**	0.00	64.03	0.00	71.09	0.00	**97.97**	0.00	79.32	0.00	77.50	0.00	91.48	0.00
w/o DF	97.40	-1.05	94.01	-1.03	61.66	-2.36	70.87	-0.22	97.93	-0.04	**81.51**	2.19	75.82	-1.68	89.76	-1.72
w/o KG	98.54	0.09	94.29	-0.75	62.82	-1.20	70.06	-1.03	96.30	-1.67	76.77	-2.56	77.39	-0.10	**91.51**	0.03
w/o DF KG	97.09	-1.36	93.79	-1.25	58.62	-5.40	70.01	-1.09	96.21	-1.76	78.49	-0.83	75.20	-2.29	89.96	-1.52
KDF+TABBIE	**98.93**	0.00	94.99	0.00	**65.49**	0.00	**71.74**	0.00	97.33	0.00	78.74	0.00	**79.01**	0.00	90.58	0.00
w/o DF	97.89	-1.04	94.78	-0.21	62.64	-2.85	71.60	-0.14	97.34	0.01	80.29	1.55	78.08	-0.93	88.63	-1.95
w/o KG	98.90	-0.04	94.87	-0.11	63.33	-2.16	71.76	0.02	95.38	-1.96	76.35	-2.39	77.99	-1.02	90.71	0.14
w/o DF KG	97.49	-1.45	94.42	-0.56	62.36	-3.13	71.16	-0.58	95.34	-1.99	77.39	-1.36	77.70	-1.32	88.43	-2.15

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
9

☑ A2. Did you discuss any potential risks of your work?
10

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
10

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
C

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
E

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
E

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
E

D ☑ **Did you use human annotators (e.g., crowdworkers) or research with human participants?**
3

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
C

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
C

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
3

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
3

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
C