Zero-shot Key Information Extraction from Mixed-Style Tables: Pre-training on Wikipedia

Qingping Yang[1,2], Yingpeng Hu[1,2], Rongyu Cao[1,2], Hongwei Li[3], Ping Luo[1,2,4]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences,
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Research Department, P.A.I. Ltd., Beijing 100025, China
[4]Peng Cheng Laboratory, Shenzhen 518066, China
{yangqingping17s, huyingpeng18s, caorongyu, luop}@ict.ac.cn, lihw@paodingai.com

***Abstract*—Table, widely used in documents from various vertical domains, is a compact representation of data. There is always some strong demand to automatically extract key information from tables for further analysis. In addition, the set of keys that need to be extracted information is usually time-varying, which arises the issue of zero-shot keys in this situation. To increase the efficiency of these knowledge workers, in this study we aim to extract the values of a given set of keys from tables. Previous table-related studies mainly focus on relational, entity, and matrix tables. However, their methods fail on *mixed-style* tables, in which table headers might exist in any non-merged or merged cell, and the spatial relationships between headers and corresponding values are diverse. Here, we address this problem while taking mixed-style tables into account. To this end, we propose an end-to-end neural-based model, called Information Extraction in Mixed-style Table (IEMT). IEMT first uses BERT to extract textual semantics of the given key and the words in each cell. Then, it uses multi-layer CNN to capture the spatial and textual interactions among adjacent cells. Furthermore, to improve the accuracy on zero-shot keys, we pre-train IEMT on a dataset constructed on 0.4 million tables from Wikipedia and 140 million triplets from Ownthink. Experiments with the fine-tuning step on 26,869 financial tables show that the proposed model achieves 0.9323 accuracy for zero-shot keys, obtaining more than 8% increase compared with the model without pre-training.**

***Index Terms*—Mixed-style table, information extraction**

I. Introduction

Table, an intuitive and easy-to-use tool for efficiently organizing, presenting a collection of facts, is widely used on the Web and in enterprises. There is always some strong demand to extract key information from tables for further analysis. For example, in the financial field, tables are used to disclose material information of company's business, financial statements, biographies of officers and directors, and detailed information about compensation and litigation. Therefore, financial practitioners spend a lot of time and burdensome labor in collecting and integrating key information from massive tables in financial documents of listed companies, then analyze the listed companies based on that information. It creates an opportunity for automatically extracting key information from tables with low human effort.

In this paper, we focus on *Key Information Extraction (KIE)* from tables, namely taking a key and a table as input and outputting a cell from the table containing the corresponding value, which output cell is called *Cell of Interest (CoI)*. An example of table key information extraction is illustrated in Figure 1. As shown, given the key "Investment capital of forward foreign exchange" and the table, our goal is to extract the cell that contains the corresponding value "63,776,900" in the green box. KIE from tables is the vital step to support many downstream applications, such as knowledge base construction [1], table retrieval [2], table understanding [3]. To the best of our knowledge, this paper is the first work to explore zero-shot KIE from mixed-style tables.

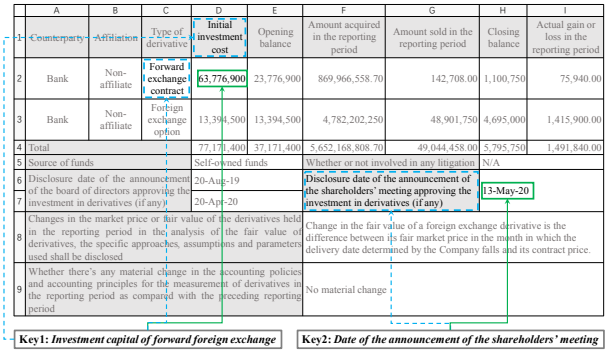

Fig. 1. An example of KIE from tables. The table is translated from a Chinese annual report for clarity. The cells in gray color are table headers. The text below the table are the given keys. The cells in green boxes and blue boxes represent the CoIs and the corresponding trigger cells, respectively.

The amount of all the keys needed to consider might be indeed massive on the task of KIE from tables, while involving the issue of zero-shot keys. Several attempts have been made to tackle KIE from the invoice or receipts [4], [5] where only single-digit keys need to extract (e.g. 4 fields in SROIE [6]). However, KIE from tables may involve hundreds or thousands of keys, since semi-structured tables are endowed with powerful information presentation capabilities and are used deeply in various domains, especially in finance. Thus, on the one hand, labeling large-scale training data for each key is both labor-intensive and unscalable. On the other hand, even after labeling all preset keys, keys that have not been encountered before may still appear in subsequent applications over time, which we call *zero-shot keys*. To enhance the performance on

Name	Ray Stark
Age	16
Gender	Female
Birthplace	Winterfell
Profession	assassin

(a) Entity Table

Name	Gender	Age
Jon Snow	Male	22
Arya Stark	Female	16
Tyrion Lannister	Male	32
Daenerys Targaryen	Female	21

(b) Relational Table

Item	In 2019	In 2018	In 2017
Total assets	39,638.00	26,761.05	22,304.23
Owners' equity attributable to the parent company	27,560.07	21,315.64	12,794.71
Asset-liability ratio (parent company)(%)	11.76	19.13	39.11
Operating income	24,098.90	25,619.01	23,379.00
Net profit	8,158.42	5,473.73	9,325.76

(c) Matrix Table

	A	B	C	D	E	F	G	H	I
1	Counterparty	Affiliation	Type of derivative	Initial investment cost	Opening balance	Amount acquired in the reporting period	Amount sold in the reporting period	Closing balance	Actual gain or loss in the reporting period
2	Bank	Non-affiliate	Forward exchange contract	63,776,900	23,776,900	869,966,558.70	142,708.00	1,100,750	75,940.00
3	Bank	Non-affiliate	Foreign exchange option	13,394,500	13,394,500	4,782,202,250	48,901,750	4,695,000	1,415,900.00
4	Total			77,171,400	37,171,400	5,652,168,808.70	49,044,458.00	5,795,750	1,491,840.00
5	Source of funds			Self-owned funds		Whether or not involved in any litigation		N/A	
6	Disclosure date of the announcement of the board of directors approving the investment in derivatives (if any)			20-Aug-19		Disclosure date of the announcement of the shareholders' meeting approving the investment in derivatives (if any)		13-May-20	
7				20-Apr-20					
8	Changes in the market price or fair value of the derivatives held in the reporting period in the analysis of the fair value of derivatives, the specific approaches, assumptions and parameters used shall be disclosed					Change in the fair value of a foreign exchange derivative is the difference between its fair market price in the month in which the delivery date determined by the Company falls and its contract price.			
9	Whether there's any material change in the accounting policies and accounting principles for the measurement of derivatives in the reporting period as compared with the preceding reporting period					No material change			

(d) Mixed Table

Fig. 2. Examples of relational, entity, matrix and mixed tables.

zero-shot keys, we need to capture the semantics of keys and improve the model generalization.

For a given key, there are some critical phrases that act as cues to pinpointing the information to be extracted. We call such phrases *triggers*, which can be regarded as a necessary and sufficient cue to recognize its corresponding value even if we mask the value. We call the cell that contains trigger as *trigger cell*. For example, for the Key2 "Date of the announcement of the shareholders' meeting " in Figure 1, the CoI is cell `H6`, which is pinpointed by the trigger cell `F6`. However, labeling trigger cells will strengthen the burden of the annotators and increase the cost. Hence, we consider building a straightforward yet effective model, which extracts CoIs from diverse styles of tables in an end-to-end way.

Different styles of tables bring varying challenges for KIE from tables. One of the most prominent features of tables is the diversity of the table structures and layouts [7]. We extend the table taxonomy from [8] with *mixed-style tables*, as shown in Figure 2 which introduces four main classes of tables: entity tables, relational tables, matrix tables and mixed-style tables. This table taxonomy is based on the position and orientation of headers and the alignment of data cells, since the *header* of a table mainly presents the attribute labels for the data region, which are the key factors that determine the layout style of the table [9], [10]. A mixed table can usually be divided into several other tables, which may have different types. For example, in Figure 2(d), the sub-table `[A1:I3]` is a relational table, while the sub-table `[A5:E7]`, `[F5:I7]` and `[A8:I9]` are entity tables. Comparing with the other three types of tables, mixed tables have a more complicated layout style, in which the headers and data cells are arranged disorderly in the table structure. As shown in Figure 2(d), the header `F5,F6` appear in the center of the table, separated from other headers. Besides, during the editing of the mixed table, the editor merges cells freely to achieve his expected layout. In Figure 2(d), the header `A5` spans 3 columns and the header `A8` spans 5 columns, which makes the table layout structure complicated.

How to parse the semantics of diverse styles of tables and extract key information from them still remains a major challenge. Existing studies related to KIE from tables [11], [12] mostly require relatively fixed table headers to identify table content, therefore they focus on relational tables and entity tables. However, matrix tables and mixed tables play a more important role, especially in the financial sector, since they have a powerful capability to present much richer information than relational tables and entity tables. According to the empirical study on public financial disclosure documents of our dataset, the proportion of matrix tables and mixed tables are higher than 90%. In this study, we explore less investigated matrix tables and mixed tables on the task of KIE. For such tables, the flexibility of the table layout leads to great uncertainty in where the trigger cells and CoIs will appear. Instead of directly parsing complicated table structures, our proposed method applies an end-to-end framework to address KIE from tables which takes all the four table types we introduced into account.

To address this task, we propose an end-to-end KIE model, called Information Extraction from Mixed-style Table (IEMT), whose design rests on a few observations on how key information is often laid out in diverse styles of tables (see Section II). For each cell, we input its text sequence to BERT [13] to obtain its textual representation, and concatenate it with the textual representation of the key. We use a multi-layer CNN [14] to capture the interaction feature of each cell and arrange 0/1 classification over each cell in the table. To address the issue of zero-shot keys, we construct a dataset of the tables on Chinese Wikipedia (https://zh.wikipedia.org) and Ownthink (https://www.ownthink.com/knowledge.html) and pre-train the model on it to improve the performance on zero-shot keys.

Based on a target financial dataset that contains 26,869 tables, the proposed IEMT model obtains an accuracy of 0.9255 for extracting zero-shot keys. Comparing with the model training from scratch, the result of fine-tuning on the pre-trained model obtains 0.0752 accuracy improvement. Furthermore, ablation studies demonstrate that each module of IEMT obtains a prominent improvement of accuracy. Interestingly, empirical experiments show that IEMT has the capacity to recognize the true trigger cells and highly depends on the trigger cell to pinpoint the CoI, even though the trigger cells are unsupervised during training.

II. Observations on KIE from Tables

In this section, we introduce four critical observations about this task that inform our design.

Observation 1 *The layout of tables in financial documents is not explicitly available.* Since there is no information about table headers, the types of the table are hard to identify. Although the position of the headers is relatively fixed in relational tables, entity tables, and matrix tables, the position of the headers is variable in mixed tables. Thus, the trigger cell and the CoI might exist in any merged or non-merged cell in

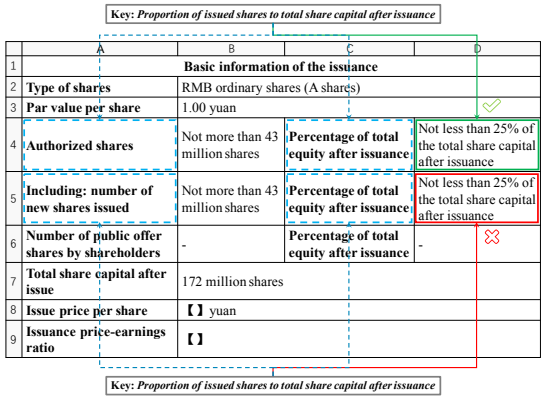

Fig. 3. Long distance dependencies between trigger cells and CoIs.

the given table when we cannot determine the layout style of the table. For example, in Figure 1, the trigger of key "Disclosure date of the announcement of the shareholder's meeting" locates at cell F6 of the table. However, this phenomenon will not happen in relational, entity, or matrix tables.

Observation 2 *The criterion for matching the key and the trigger is the semantic similarity.* For a given key, its corresponding triggers in different tables may have distinct expressions but the same meaning. For example, the corresponding trigger of the key "Investment capital of forward foreign exchange" could be "The initial investment cost of forward exchange contract", "The initial investment amount of forward foreign exchange", "The capital of forward foreign exchange" or other expressions. Thus, the model should recognize the trigger cell according to the semantic similarity between the key and the table cells instead of the textual similarity. In addition, the expressions of a trigger for a given key might be scattered into several trigger cells. In Figure 1, "Forward exchange contract" and "Initial investment cost" in the blue box together consist of the trigger of the key "Investment capital of forward exchange" in the table.

Observation 3 *Long distance dependencies may occur between trigger cells and CoIs.* As we mentioned above, the number of trigger cells can be more than one for a given key. Some of the trigger cells may be far away from the cell. For example in Figure 3, the given key is "Proportion of issued shares to total share capital after issuance", and the correct CoI is D4 (green box in Figure 3). If we only consider the CoI and its closet trigger cell, the wrong CoI D5 (red box in Figure 3) might be predicted, because in such a perspective, cell D4 and D5 are the same. However, the correct CoI D4 is dominated by trigger A4 and C4, in which A4 is far from D4.

These observations are common in tables and inspire the design of our IEMT which detailed in the next section.

III. Model

In this section, we first formulate the problem of table key information extraction in Section III-A, then introduce the details of each module in IEMT in Section III-B. Finally, we present the procedure of pre-training in Section III-C.

A. Problem Formulation

In this section, we formulate the problem of table key information extraction. Given a key and a table, our goal is to extract the CoI in the table.

Formally, we denote the set of non-zero-shot keys and zero-shot keys as $\mathcal{K}_n = \{k_i\}_{i=1}^{N_n}$ and $\mathcal{K}_z = \{k_i\}_{i=1}^{N_z}$, respectively. The keys are defined by professional financial practitioners. Note that $K_n \cap K_z = \emptyset$. A key k_i is a natural language phrase composed of a sequence of words. Then, we define the labeled training set as a collection of 3-tuples, $\mathcal{D}_{tr} = \{(k_i, T_i, c_i^*)|k_i \in \mathcal{K}_n\}_{i=1}^{N_{tr}}$ and the labeled test set as a collection of 3-tuples, $\mathcal{D}_{te} = \{(k_i, T_i, c_i^*)|k_i \in \mathcal{K}_n \cup \mathcal{K}_z\}_{i=1}^{N_{te}}$. Here, k, T and c^* denote the key, table structure and ground-truth CoI, respectively. A table T is a two-dimensional layout of cells, thus we denote it with a set of cells, $T = \{c_{\langle i,j\rangle}\}_{i,j=1,1}^{m,n}$, where m and n represents the number of rows and columns in this table. $c_{\langle i,j\rangle}$ denote the cell in the j-th column and the i-th row in T. Then, we define $P(c_{\langle i,j\rangle}|k_i, T_i)$ as the probability of being the CoI of the cell $c_{\langle i,j\rangle}$.

The problem now is to learn a model trained with the training set $\mathcal{D}_{tr}$, but can still extract the CoI in the test set $\mathcal{D}_{te}$, no matter whether the key is zero-shot or non-zero-shot. Driven by Observation 1 in Section II, for each data (k, T) in the test set, our model seeks the CoI $\bar{c}$ with the highest probability over all cells in the given table.

B. IEMT: Value Extraction in Mixed-style Table

In this section, we introduce the framework of the proposed IEMT model, which consists of two modules: an *encoder* and a *scorer*. The architecture of the proposed IEMT is depicted in Figure 4.

1) Encoder: Recall Observation 2 in Section II, it is critical to capture the textual semantics of the key and each cell of the table. Therefore, we use BERT [13] to encode the key and the text sequence in each cell. BERT is pre-trained on large-scale text for multiple downstream NLP tasks. Therefore, it provides a powerful context-dependent representation, which contains the semantic information to some extent, for each text sequence.

The encoder takes as input a given key k and a table T, and outputs a table tensor v (the cyan tensor in Figure 4). The text sequence in each cell is independently converted into a d-dimensional vector with BERT, thus table T is converted into a tensor with the size of (m, n, d). Especially, if a merged cell spans several rows or columns, these sub-cells will share the same vector. By the way, we add a [CLS] token at the beginning of the text sequence as BERT does to capture the overall feature of the sequence. To eliminate the effect of variable table size, we normalize the row and column index into the range $[-1, 1]$ as additional two dimensions to the table tensor, which enable the model to perceive the relative positional information of cells in table T. We further concatenate the key vector with each cell vector to combine the information of the given key with the table. Thus, the table tensor is with the size of (m, n, $2d$+2) and denoted with v.

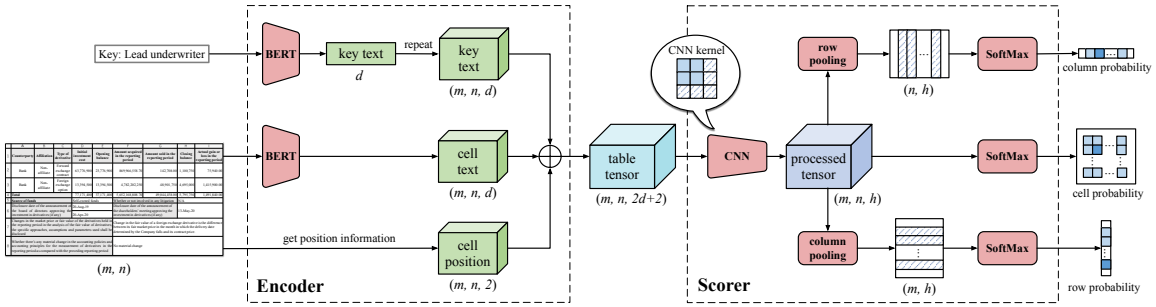

Fig. 4. The framework of IEMT.

2) Scorer: To integrate the relational information between each cell with the surrounding cells, we adopt a multi-layer CNN, which takes as input table tensor v with the size of $(m, n, 2d + 2)$ and outputs a processed tensor (the blue tensor in Figure 4) with the size of (m, n, h). There are 6 convolutional layers with batch normalization [15] in our CNN model Specifically, we mask the right and bottom part of the convolution kernels in the network with zero, because we only need to focus on the left and top parts of CoIs. Then, for each cell in the table, a Sigmoid layer is used to obtain the probability of being the CoI, .

As the primary goal of learning, we want to minimize the error of each cell's prediction score. Denote the label of cell c as $l^c \in \{0, 1\}$. The cell objective function is as follows:

$$L_{cell} = -\sum_{c \in T} [l^c \log(P(c)) + (1 - l^c) \log(1 - P(c))], \quad (1)$$

where $P(c)$ refers to $P(c_{\langle i,j \rangle}|k, T)$.

3) Joint Learning Objective: To make the model can capture the relational features between the cells with long distance, we design two auxiliary learning objectives. Recall that the processed tensor (the blue tensor in Figure 4) after multi-layer CNNs. We use max-pooling on each row and column, and get each row's vector and each column's vector. Finally, we use a Sigmoid layer to obtain the probability of being the row or column that contains the CoI.

Formally, we denote the i-th row vectors as r_i, l_i^r equals to 1 if the CoI locates at the i-th row otherwise 0. The auxiliary row objective is calculated as follows:

$$L_{row} = -\sum_{i=1}^{n} [l_i^r \log(P(r_i)) + (1 - l_i^r) \log(1 - P(r_i))], \quad (2)$$

L_{col} is calculated similarly to L_{row}. Thus, the joint learning objective is designed as: $L = L_{cell} + \alpha(L_{row} + L_{col})$.

C. Pre-training

To enhance the generalization ability of the model, we build a pre-training dataset, called Wiki Dataset, with Ownthink and the tables on Chinese Wikipedia, and pre-train the proposed model on this dataset. To obtain the keys and values in tables on Wikipedia, we consult the (entity, attribute, value)-tuples in Ownthink, which is a huge Chinese knowledge graph that contains about 140 million tuples. Here, the entities and attributes can be used as the keys and the values can be used as the values in our table KIE task.

TABLE I
THE DETAILED FINANCIAL DATASET AND WIKI DATASET STATISTICS.

	Financial Dataset	Wiki Dataset
#Page	368,004	133,552
#Table	26,869	432,340
#Key	732	250,186
#Cell (Average) in a table	64.63	78.51
#Row (Average) in a table	11.59	12.85
#Column (Average) in a table	5.07	5.59

In detail, Chinese Wikipedia only provides a large number of tables and Ownthink only provides a large number of (entity, attribute, value)-tuples, then the key point is how to link these two independent corpus. To this end, for each tuple in Ownthink, we traverse each cell of every tables in Wikipedia and match it with entity, attribute and value based on exact text matching. Then, we only retain the tuple where the entity, attribute and value matches the same table simultaneously, meanwhile, the entity cell, attribute and value locates at the same row. Finally, since all the tables we collected are relational tables, we further expand the more pieces of tuples by scanning the relational table from the top to the bottom. Note that, we concatenate the text in entity and attribute to construct the key. After constructing Wiki Dataset, we fine-tune the model on our financial table dataset.

IV. EXPERIMENTS

In this section, we first introduce the two datasets we constructed, one is the target dataset called Financial Dataset, and the other one is the pre-training dataset called Wiki Dataset. Then, we present two ways of splitting Financial Dataset into training set, validation set and test set.

Based on these datasets, we compare the proposed IEMT model with the baseline model and present some ablation experiments with discussion and analysis. Finally, we conduct some experiments to verify that our model is capable of recognizing the trigger cells implicitly.

A. Baseline

Since previous studies have not dealt with KIE from mixed-style tables, we perform KATA [5] as our baseline, which aims

TABLE II
COMPARING DIFFERENT VARIANTS OF IEMT ON THE TEST SET.

Row	Model Setting	Split Method	
		non-zero-shot split	zero-shot split
1	KATA	0.9427	0.4266
2	IEMT from scratch	0.9869	0.8505
3	IEMT	**0.9873**	**0.9323**
4	IEMT w/o joint objective	0.9766	0.8831
5	IEMT w/o masked kernel	0.9645	0.8772
6	IEMT w/o cell position	0.9801	0.9044

to extract key information from document pages. The model is extended by LayoutLM [16] with explicitly trigger-supervised training, while not applicable in our dataset. Therefore, we convert the table into a picture at first, and then use KATA to extract the texts in the CoI without KATA's auxiliary loss about triggers. This setting also demonstrates the superiority of our model without the additional annotation of the triggers.

B. Dataset

To build Financial Dataset, we download a total of 871 public PDF documents from CNINFO(http://www.cninfo.com.cn), a financial information disclosure website. Most of these documents are the annual reports and prospectuses of listed companies. We extracted tables from these documents with a table extraction and recognition tool. Also, we ask the professional financial practitioners to pre-define a set of keys (e.g. "Proposed investment in raised assets"). The information of the keys is common in financial documents. Then, we ask financial practitioners to annotate the corresponding CoIs of the keys for each table. Each table is assigned to at least two annotators for annotating the CoI. If the results for a value are different, another senior annotator will address the conflicts and output the final answer. The detailed information of Financial Dataset and Wiki Dataset is shown in Table I.

To handle the zero-shot learning problem, we design a split method called *zero-shot split*. To split the dataset under a zero-shot method, the dataset is split into 8:1:1 over different keys for training, validation, and testing. We can guarantee that each key in the test set never appears in the training set. Furthermore, to evaluate the accuracy of non-zero-shot keys, we design another split method, called *non-zero-shot split*. For each sub-dataset of a key, we split it into 8:1:1 for training, validation, and testing.

C. Implementation Details

We adopt pre-trained BERT-Base (Chinese) to encode the keys and the texts in tables. In Figure 4, d is 768 and h is 512. The hyper-parameter α is set to 0.1 (selected from 0.01, 0.1, 0.2 and 0.5). We use the gradient descent algorithm with Adam [17] to train our model. The learning rate is set to 10^{-4} and the batch size is 16. In our experiments, we leverage GPU (GeForce GTX 1080Ti) to train and infer.

D. Results and Discussion

In this section, we aim to answer these research questions:

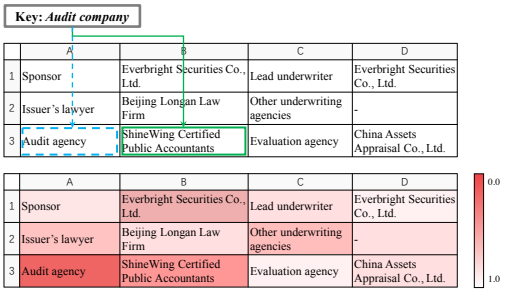

Fig. 5. An example to show the importance of each cell.

• **RQ1: What is the effectiveness of the IEMT model compared with the baseline model KATA?** we compare the accuracy of the IEMT model from scratch with the baseline model KATA. We present the results in row 1 and 2 of Table II. IEMT from scratch obtains 0.8505 accuracy under zero-shot split and 0.9869 accuracy under non-zero-shot split. While KATA obtains 0.4266 accuracy under zero-shot split and 0.9427 accuracy under non-zero-shot split. The improvement of the accuracy of IEMT compared with KATA is 0.4239 under zero-shot split and 0.0442 non-zero-shot split. Our IEMT is more effective than KATA on both the zero-shot test set and the non-zero-shot test set.

• **RQ2: What is the effectiveness of the additional modules in the IEMT model?**, we design ablation studies to evaluate the influence of each additional module in IEMT. We present the results in row 3, 4, 5, and 6 of Table II. We first replace the joint objective with only a cell objective. IEMT without joint objective obtains 0.0107 accuracy decrease under non-zero-shot split and 0.0492 decrease under zero-shot split. Then, we remove the position embedding of the cells in a table. IEMT without cells position obtains 0.0072 accuracy decrease under non-zero-shot split and 0.0279 accuracy decrease under zero-shot split. We also replace the masked convolution kernels with normal convolution kernels. IEMT without masked convolution kernels obtains 0.0228 accuracy decrease under non-zero-shot split and 0.0551 accuracy decrease under zero-shot split. In short, each module in IEMT increases the accuracy effectively.

• **RQ3: What is the accuracy order under non-zero-shot split and zero-shot split?**, we design experiments under non-zero-shot split and zero-shot split, respectively. For each row in Table II, the accuracy order under three split methods is "non-zero-shot split $\geqslant$ zero-shot split". The reason is that predicting non-zero-shot keys is easier than zero-shot keys.

E. Case Study and Limitations

As shown in Figure 5, the given key is "Audit company" and the given table has 3 rows and 4 columns. To investigate the effect of each cell for extracting the CoI `B3`, for each cell, we mask its text with an unknown token `[UNK]` and use IEMT to predict the probability of the CoI `B3`. We use $1-$probability to denote the importance of the masked cell, i.e. the lower probability of the CoI, the more important the masked cell.

Then, we draw the importance of each cell with red color. We observe that the trigger cell "Audit agency" obtains the darkest color, namely the highest importance. Although the proposed IEMT can recognize the trigger cell implicitly while extracting the CoI, it cannot extract the exact trigger cell for a given key.

V. RELATED WORK

Recently, significant studies have focused on the work about key value extraction from tables [11], [18], plain texts [19] and documents [20], [21]. In this paper we focus on key value extraction from mixed-style tables. We extract the only one corresponding value cell from the given table, whose structure is quite complicated. However, early table-related works only focus on structure-limited tables, such as relational tables and entity tables. Therefore, previous works on tables are not valid for mixed-style tables. In [11], they classify the tables into three types: two-column tables, relational tables, and colon-delimited pair tables. Their method is still unable to extract key values from matrix tables and mixed tables with complex styles, which are common in financial documents. In [19], they propose an approach for key value extraction via question answering using a multi-task framework. In [20], they extract ($attribute$,$value$)-pairs from Wikipedia articles with a self-supervised approach. In [21], they use representation learning to tackle the problem of extracting structured information from form-like documents. The extraction system uses knowledge of the types of the target fields to generate extraction candidates, and a neural network to learn a dense representation of each candidate based on neighboring words in the document.

Zero-shot learning is proposed by [22]. It is a promising learning paradigm, where the goal is to learn a classifier $f : X \rightarrow Y$ that must predict novel values of Y that were not omitted from the training set [22]. Recently, many works adopt zero-shot learning in various tasks, such as relation extraction, entity extraction, image recognition, etc. [22]–[25]. In [24], they focus on a zero-shot task of extracting entities from web pages. The former methods usually require seed entities and then extract the target entities that are similar to the seed entities. However, they replace the seed entities with a natural language query and predict if each candidate word is an entity or not. In [23], they aim to open-domain relation extraction from web pages. The pages are on unseen websites. To enhance the generalization ability, they propose a graph neural network model. The model encodes the semantic textual and visual patterns from different websites.

VI. CONCLUSION

In this paper, we propose the problem of table key information extraction and focus on extracting zero-shot key information from mixed-style tables in financial documents. We propose a straightforward yet effective table key information extraction model IEMT and enhance the generalization ability of the model by pre-training and fine-tuning. The experiments show that the performance of our model is outstanding, and our model can recognize the trigger cells implicitly as humans while extracting CoIs.

VII. ACKNOWLEDGMENTS

The research work was supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002104, the National Natural Science Foundation of China under Grant Nos. 62076231 and U1811461.

REFERENCES

[1] Z. Chen and M. Cafarella, "Integrating spreadsheet data via accurate and low-effort extraction," in *KDD*, 2014.
[2] S. Zhang and K. Balog, "Ad hoc table retrieval using semantic similarity," in *WWW*, 2018.
[3] R. Rastan, H.-Y. Paik, and J. Shepherd, "Texus: A unified framework for extracting and understanding tables in pdf documents," *Information Processing & Management*, 2019.
[4] B. P. Majumder, N. Potti, S. Tata, J. B. Wendt, Q. Zhao, and M. Najork, "Representation learning for information extraction from form-like documents," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6495–6504.
[5] R. Cao and P. Luo, "Extracting zero-shot structured information from form-like documents: Pretraining with keys and triggers," in *AAAI*, 2021.
[6] Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, and C. Jawahar, "Icdar2019 competition on scanned receipt ocr and information extraction," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1516–1520.
[7] K. Braunschweig, "Recovering the semantics of tabular web data," Ph.D. dissertation, Technischen Universitat Dresden, 2015.
[8] O. Lehmberg and C. Bizer, "Web table column categorisation and profiling," in *Proceedings of the 19th International Workshop on Web and Databases*, 2016, pp. 1–7.
[9] J. Fang, P. Mitra, Z. Tang, and C. L. Giles, "Table header detection and classification," in *AAAI*, 2012.
[10] S. Seth and G. Nagy, "Segmenting tables via indexing of value cells by table headers," in *ICDAR*, 2013.
[11] Y. W. Wong, D. Widdows, T. Lokovic, and K. Nigam, "Scalable attribute-value extraction from semi-structured text," in *ICDM*, 2009.
[12] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. M. Eisenschlos, "TAPAS: Weakly supervised table parsing via pre-training," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4320–4333.
[13] D. Jacob, C. MingWei, L. Kenton, and T. Kristina, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.
[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.
[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.
[16] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1192–1200.
[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
[18] S. Zhang and K. Balog, "Web table extraction, retrieval and augmentation: A survey," *ACM Transactions on Intelligent Systems and Technology*, 2020.
[19] G. Zheng, S. Mukherjee, X. L. Dong, and F. Li, "Opentag: Open attribute value extraction from product profiles," in *KDD*, 2018.
[20] B. P. Majumder, N. Potti, S. Tata, J. B. Wendt, Q. Zhao, and M. Najork, "Representation learning for information extraction from form-like documents," in *ACL*, 2020.
[21] W. C. Brandao, E. S. Moura, A. S. Silva, and N. Ziviani, "A self-supervised approach for extraction of attribute-value pairs from wikipedia articles," in *SPIRE*, 2010.
[22] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *NeurIPS*, 2009.
[23] C. Lockard, P. Shiralkar, X. L. Dong, and H. Hajishirzi, "Zeroshotceres: Zero-shot relation extraction from semi-structured webpages," in *ACL*, 2020.
[24] P. Pasupat and P. Liang, "Zero-shot entity extraction from web pages," in *ACL*, 2014.
[25] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata, "Feature generating networks for zero-shot learning," in *CVPR*, 2018.