SHEETPT: Spreadsheet Pre-training Based on Hierarchical Attention Network

Ran Jia[1*†], Qiyu Li[2*‡], Zihan Xu[2‡], Xiaoyuan Jin[2‡], Lun Du[1], Haoyu Dong[1], Xiao Lv[1], Shi Han[1], Dongmei Zhang [1]

[1] Microsoft Research Asia
[2] Peking University
{raji, lun.du, hadong, xilv, shihan, dongmeiz}@microsoft.com, {liqiyu0728, xzhpku, jinxy}@pku.edu.cn

Abstract

Spreadsheets are an important and unique type of business document for data storage, analysis and presentation. The distinction between spreadsheets and most other types of digital documents lies in that spreadsheets provide users with high flexibility of data organization on the grid. Existing related techniques mainly focus on the tabular data and are incompetent in understanding the entire sheet. On the one hand, spreadsheets have no explicit separation across tabular data and other information, leaving a gap for the deployment of such techniques. On the other hand, pervasive data dependence and semantic relations across the sheet require comprehensive modeling of all the information rather than only the tables. In this paper, we propose SHEETPT, the first pre-training technique on spreadsheets to enable effective representation learning under this scenario. For computational effectiveness and efficiency, we propose the coherent chunk, an intermediate semantic unit of sheet structure; and we accordingly devise a hierarchical attention-based architecture to capture contextual information across different structural granularities. Three pre-training objectives are also designed to ensure sufficient training against millions of spreadsheets. Two representative downstream tasks, formula prediction and sheet structure recognition are utilized to evaluate its capability and the prominent results reveal its superiority over existing state-of-the-art methods.

Introduction

With the superior ability of efficient data management and presentation, spreadsheets are widely used by billions of users in public and internal data storage and analysis. They are becoming a critical scenario of document intelligence, where advanced intelligence techniques can improve user effectiveness and experience such as formula and chart recommendation during analysis. To accomplish these goals, understanding spreadsheets from both structure and semantics is an important foundation and has attracted attention from researchers in recent years.

Most existing studies on spreadsheets take a single table as the input, such as error detection (Huang and He 2018; Zhang et al. 2021), formula prediction (Chen et al. 2021; Cheng et al. 2021) and chart recommendation (Zhou et al. 2021). Some (Huang and He 2018; Zhou et al. 2021; Chen et al. 2021) of them only discuss relational tables with simple and definitive schema, and others (Zhang et al. 2021; Cheng et al. 2021) require explicit table boundary and structural information like the position of headers. Such designs heavily rely on techniques that detect table boundary and structure, and errors from preceding models will accumulate and impact performance on downstream tasks. Essentially, these work regard spreadsheets as a type of tabular data, which neglects the fact that they are a kind of document where data is organized on the grid with more abundant analytical components like formulas and charts.

As a unique type of document, spreadsheets are distinct from tables in other documents in the following aspects. 1) Table boundary in spreadsheets is usually not explicit and is even difficult to be identified on some sheets based on the general table definition; 2) Even if the table boundary can be determined, other cells outside the table also contain valuable information, such as the data description and comments; 3) Users will process and analyze across different tables, such as creating cross-table formulas and charts. For convenience, there can be multiple tables with semantic and calculation dependence in one sheet. Hence, only considering tables in spreadsheets will lead to significant information loss, and a more comprehensive spreadsheet modeling method is in urgent need.

To bridge the gap between tables and spreadsheets, we desire an approach that can directly handle the entire sheet, which means the sheet is not divided into multiple tables and separately modeled, instead all the cells are input together and the contextual relations among cells can be holistically captured. Due to the high complexity of spreadsheets data, data labeling for various tasks is extremely hard, so we turn to leveraging the pre-training technique. But it still faces some challenges:

***Challenge 1*: Large input size**. Spreadsheets usually consist of numerous cells and the entire token sequence can be extremely long. According to our statistics on more than 400K sheets, the 85^{th} percentile of the number of cells is about 4K, and the length of the corresponding token sequence is more than 13K, which greatly exceeds the usual token limit for transformer-based models and raises challenges to both

[*] These authors contributed equally.
[†] Corresponding author.
[‡] Work done during internship at Microsoft Research Asia.

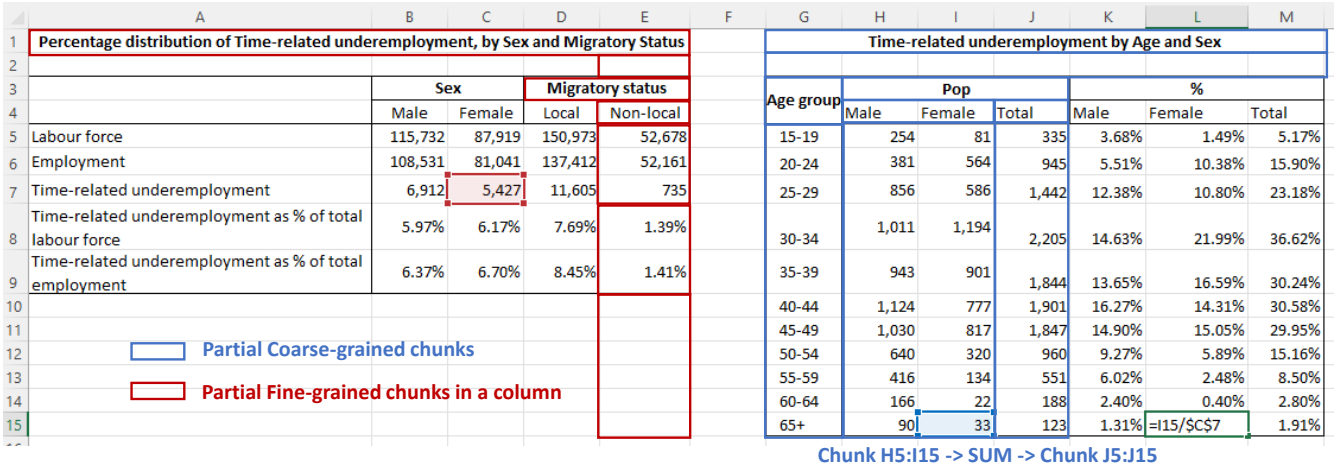

Figure 1: A typical example of a spreadsheet with two tables. There are semantic and calculation relations between the data of the two tables. We present some divisions of coherent chunks on the sheet. Some chunks have aggregation relations represented by formulas, such as chunk H5:I15 and chunk J5:J15.

GPU overhead and semantic extraction.

***Challenge 2*: Sparse semantics distribution**. In most cases, the semantic information of a sheet locates in header cells since they denote the semantics of the corresponding data values. Typically, data cells account for most of a spreadsheet and there are only a small number of header cells (around 20% based on our statistics on a labeled dataset). So the semantics of the spreadsheets are sparsely distributed and can only be found on a small part of cells, increasing the difficulty in capturing semantics without known cell types.

***Challenge 3*: Long-range dependence**. In spreadsheets, a cell is not only correlated to the cells nearby or in the same row or column, but also can have intricate relations with distant cells. For instance, cell L15 in Figure 1 has an implicit semantic dependence on its header cells. What's more, there is a calculation dependence between L15, I15, and C7 explicitly marked by the formula. There are quite a few cells with long-range dependence, which poses a tough challenge for capturing such semantics.

To address these challenges, we propose **SHEETPT**, a novel pre-training architecture based on hierarchical attention network to model the entire sheet in this paper. Specifically, We design an intermediate data granularity called coherent chunk and describe the hierarchical structure of spreadsheets with token, cell, and chunk levels. A multi-head hierarchical attention network then captures contextual information and learns semantic representations for spreadsheets. Three categories of pre-training objectives are designed to facilitate SHEETPT, including objectives to capture local context, understand sheet structure, and learn analytical semantics. Pre-trained on millions of data, SHEETPT can serve as a foundation model of spreadsheets and empower a variety of downstream tasks. Our major contributions are summarized as follows:

- To the best of our knowledge, SHEETPT is the first attempt in large-scale spreadsheet pre-training that can handle spreadsheets with plenty of cells and alleviate sparse semantics and long-range dependence problems.
- Regarding particularity of spreadsheets, we design three categories of pre-training objectives to better learn representations of spreadsheets from massive unlabeled data.
- We create the largest spreadsheets dataset, SheetSem, labeled with detailed cell semantic types for a downstream task sheet structure recognition.
- We evaluate model performance on the downstream formula prediction task and sheet structure recognition task. Experiments indicate its superiority over baselines, especially under cross-table scenarios.

SHEETPT Model

The architecture of SHEETPT is illustrated in Figure 2 and we detail its backbone in this section. We propose a novel semantic unit, **Coherent Chunk**, to organize homogeneous cells as an intermediate data granularity in spreadsheets. Accordingly, we elaborate a novel pre-training architecture with multi-head hierarchical attention to extract contextual semantics of spreadsheets. Attention over coherent chunks is both efficient and effective to address the large input size and intricate dependence in spreadsheets. In addition, we adopt one analytical signal, the formula in spreadsheets, and devise Chunk-Level Relational Attention (CLRA) to integrate formulas and provide richer relational information.

Coherent Chunk

Unlike reading text documents, users usually do not pay attention to the content of each cell or go through them one by one, especially for massive homogeneous data cells with less semantic information. Instead, they are more interested in the header cells with informative semantics. We find that these homogeneous data cells are (1) mostly numeric; (2) usually located adjacent to each other and will form a regular (rectangular in most cases) area. It motivates us to regard cells within such a region as a whole for spreadsheet modeling. Such a region is called a "Coherent Chunk" and it is defined as a rectangular area in spreadsheets whose cells are

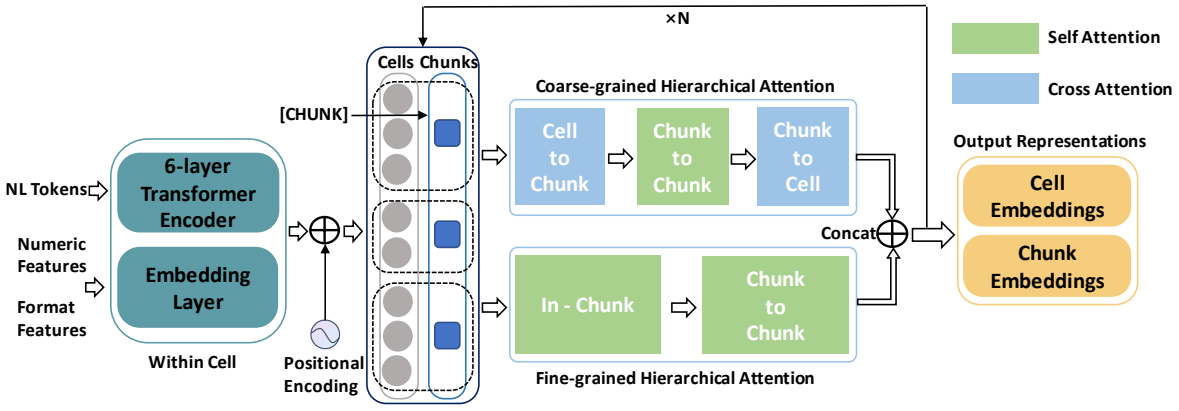

Figure 2: An overview of model architecture.

semantically homogeneous. For example, it can be data values with the same measure like the numeric cells in H5:I15 in Figure 1, or cells sharing a common semantic concept like "Male" and "Female" in H4:I4 of Figure 1.

By this design, to model spreadsheets data, there are semantic units at three levels: cell $\rightarrow$ coherent chunk $\rightarrow$ spreadsheet. On one hand, an intermediate data granularity in spreadsheets can significantly reduce the semantic units involved in the computation (from the number of cells to the number of coherent chunks). On the other hand, the coherent chunk can aggregate homogeneous cells; and inputs organized as chunks will be more intensive so that the sparse semantics issues can be alleviated. Besides, the interactions between chunks in our model are able to enrich the contextual information a single cell can obtain. In more detail, chunks serve as an intermediary to gather information from all other chunks and then pass it to each internal cell. The expansion of the receptive field for each cell is beneficial to capture the long-range dependence.

Although coherent chunks offer a bunch of benefits to the pre-training model, *how to determine coherent chunks in a spreadsheet* becomes a derivative problem. Therefore, we devise an effective heuristic approach to segment a spreadsheet into coherent chunks. The approach is based on some well-designed heuristic rules to determine whether two adjacent cells should be grouped into the same chunk by looking at the number format codes (Microsoft 2022b), prefixes and suffixes of text strings, formulas and so on.

Multi-Grained Hierarchical Attention Network

To learn representations at the token, cell, and chunk levels, we propose a novel hierarchical attention architecture and will detail it in this section.

Cell Embedding Layer In spreadsheets, there is rich information to identify cells, including text, values, positions, and formats. Accordingly, we follow (Wang et al. 2021) to encode cell semantics in embedding E_{SMT}, numerical features in E_{NUM}, position features in E_{POS} and formatting features in E_{FORMAT}.

Specifically, information of the text token sequence in each cell is encoded in the `[CLS]` token by feeding the sequence into a 6-layer transformer encoder. The encoder is initialized by DistillBERT (Sanh et al. 2019) to obtain the cell semantic embedding E_{SMT}. When there is a numerical value in the cell, we define five discrete numerical features: sign, magnitude, precision, the first and the last digit, linearize each feature and then concatenate them as the numeric embedding E_{NUM}. The position of a cell is presented as its row and column in the sheet, so we encode the coordinates of the row and column into the position embedding E_{POS}. Rich formatting features are also incorporated into the formatting embedding E_{FORMAT} by linearizing the predefined discrete formatting features, including number formats, cell borders, font styles, indents, background colors, and alignments. Eventually, the cell embeddings E_{CELL} aggregate information from the aforementioned four perspectives by a summation operation:

$$E_{\text{CELL}} = E_{\text{SMT}} + E_{\text{POS}} + E_{\text{NUM}} + E_{\text{FORMAT}}. \quad (1)$$

Hierarchical Attention As we mentioned, there can be thousands of cells in a sheet, raising the difficulty in conducting self-attention on a sequence of all the cells. So we elaborate a hierarchical attention network over the coherent chunks to incorporate self-attention. In general, cell representations are expected to incorporate contextual information from both coarse-grained and fine-grained levels. Coarse-grained information contains sketchy characteristics of cell ranges, like H5:J15 and K5:M15 in Figure 1 are the range of integers and percents, respectively. While fine-grained information includes detailed semantics between cells, I15 representing the number of the female population over the age of 65 is a typical example. Hence, as shown in Figure 2, multi-head hierarchical attention at two granularity levels, i.e., coarse-grained attention and fine-grained attention, are devised to extract thorough semantics efficiently.

Coarse-grained Attention Coarse-grained attention aims to avoid inter-cell calculation over homogeneous cells and capture the sketchy range features instead. The receptive field of each individual cell can then be broadened for a global view of the entire spreadsheet. Figure 2 demonstrates the structure as a self-attention sandwiched between two cross-

attention layers. Cross-attention between cells and the superior chunk is to summarize the local information of a chunk. Self-attention among chunks then propagates information of different regions in the spreadsheet to capture global contexts. Such signals are then delivered back to update each cell via the chunk-to-cell cross-attention. By this means, the computation cost is significantly reduced without inter-cell attention and it is no longer challenging to handle large-scale spreadsheets data.

While coarse-grained attention can tackle the problem of large input size, contextual information retrieved back from chunks is not delicate enough. For example, we may infer cell I15 in Figure 1 as an unemployment value, but it is hard to determine if it is for "Male" or "Female". This phenomenon requires a more fine-grained design for detailed contexts.

Fine-grained Attention Fine-grained attention is devised to capture detailed semantic contexts. Commonly, cells in the same row or column are usually semantically related, so we organize cells in each row or column into fine-grained chunks. Within each chunk, self-attention over a sequence consisting of cells and the corresponding chunk can delicately extract semantics between cells. And we limit the chunk-wise self-attention to chunks in the same row or column. Computations under this design are also significantly less than that over all the spreadsheet cells, so fine-grained attention can provide more informative cell embeddings while still maintaining computation efficiency.

Chunk-Level Relational Attention

Data analysis is the most major scenario when using spreadsheets; hence there are abundant analytical signals in spreadsheets, such as formulas, charts, pivot tables, and conditional formatting. These signals present additional relations among cells, and incorporating such relations can undoubtedly promote model ability of spreadsheets understanding. As a unique component of spreadsheets, the formula explicitly marks the calculation relation between cells. For instance, the aggregation formula, a representative formula type, applies aggregation functions (e.g., SUM, AVERAGE) to a range of cells, and the adjacent reference positions of cells with the same formula can indicate the relation between two chunks. In Figure 1, the relation between chunk J5:J15 and H5:I15 can be denoted as the "SUM" aggregation function. So we incorporate the intrinsic relations between chunks derived from formulas into the attention to enhance the interactions of involved chunks.

There are 19 aggregation functions in Excel (Microsoft 2022a). And we additionally include `SELF` and `EMPTY` relation to the relation vocabulary to represent self-reference and absence in formulas, respectively. Each relation in the vocabulary is assigned with a learnable relation embedding $R \in \mathbb{R}^{d_{Rel}}$ with d_{Rel} dimension. Similar to relative positional embedding (Shaw, Uszkoreit, and Vaswani 2018), Chunk-Level Relational Attention (CLRA) can be written as

$$\text{CLRA}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top + QR^\top}{\sqrt{d_{Rel}}}\right)V. \quad (2)$$

Pre-training Objectives

Pre-training objective is another critical factor for the success of the pre-training model. Therefore, we devise multiple pre-training objectives to enhance the model from different aspects of spreadsheets understanding. Figure 3 demonstrates examples of various pre-training objectives.

Pre-training Objective of Local Context

Inspired by the tabular pre-training (Dong et al. 2022), we utilize **Masked Language Modeling (MLM)** (Devlin et al. 2018) and **Cell-Level Cloze (CLC)** (Wang et al. 2021) to learn the local context at token-level and cell-level, respectively. We follow the setup of (Wang et al. 2021) and randomly mask tokens and cells, and $\mathcal{L}_{\text{MLM}}$ and $\mathcal{L}_{\text{CLC}}$ are the corresponding cross-entropy loss for semantic units at these two levels.

Pre-training Objective of Sheet Structure

SHEETPT is expected to implicitly integrate sheet structure including table boundary and header cell position into cell embeddings. So we elaborate a series of **Semantic Role Classification (SRC)** pre-training objectives to identify the semantic roles of cells and chunks. The pseudo semantic labels of cells and chunks are generated by the latest table detection and understanding techniques (Dong et al. 2019b,a), including `top/bottom/left/right border` and `inside/outside-table`. They all indicate the location of a chunk/cell in a table. Besides, there is an additional type of label denoting whether a cell is in the header region or not (i.e., header cell). All the semantic labels over cells and chunks are used in classification tasks with the corresponding cross-entropy loss, and the losses are then summarized as the SRC pre-training objective $\mathcal{L}_{\text{SRC}}$.

Pre-training Objective of Analytical Semantics

As aforementioned, analytical signals in spreadsheets can indeed be beneficial to extract relations among cells and provide an in-depth understanding. Apart from the design of relational attention, we also introduce a corresponding pre-training objective based on formulas to further strengthen its effect on SHEETPT.

Concretely, we follow (Chen et al. 2021; Cheng et al. 2021) to parse a formula into the sketch and reference ranges. The sketch of a formula is a sequence of formula tokens, replacing the reference cells with a special token `[RANGE]`. Two pre-training objectives **Next Formula Token Prediction (NFTP)** and **Reference Prediction (RP)** are devised accordingly. Cell embeddings concatenated with formula tokens are progressively fed into a transformer decoder to predict the next formula token at each step for NFTP. And its objective $\mathcal{L}_{\text{NFTP}}$ is denoted as a multi-class cross-entropy loss over all the formula tokens. As for RP, we follow (Chen et al. 2021; Cheng et al. 2021) to predict reference ranges based on the embeddings of cells and `[RANGE]` tokens, and accordingly calculate the cross-entropy objective $\mathcal{L}_{\text{RP}}$.

The overall pre-training objective can be formulated as

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{CLC}} + \mathcal{L}_{\text{SRC}} + \mathcal{L}_{\text{NFTP}} + \mathcal{L}_{\text{RP}}. \quad (3)$$

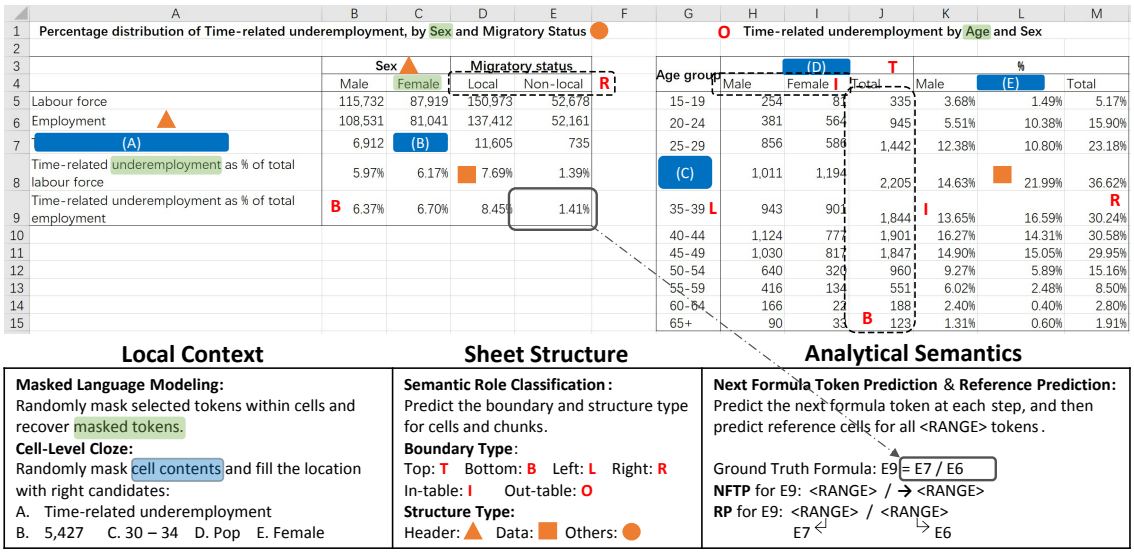

Figure 3: Examples of samples generated from a sheet for each pre-training objective. Local context objective involves MLM and CLC, two multi-class classification tasks on token vocabulary and masked candidates respectively. Sheet structure objective classifies some cells and chunks into specific boundary types and structure types. Analytical semantics objective involves NFTP, a sequence generation task over formula tokens, and RP, a classification task to predict the reference cells.

Experiments

In this section, we first describe the pre-training setup and then evaluate SHEETPT on two downstream tasks: formula prediction and sheet structure recognition.

Pre-training Setup

SHEETPT utilizes 6 layers of transformer encoder initialized with DistilBERT (Sanh et al. 2019) for processing token sequence in a cell. The number of layers of multi-grained hierarchical attention is also 6.

We crawl 16.12M public spreadsheet files from about 440K web domains. Data pre-processing includes language detection, de-duplication, and removing extremely large and small spreadsheets. Finally, we generate a pre-training dataset of 5.85M spreadsheets in English. We set the maximum number of rows and columns to 200 and 25, respectively, which covers more than 85% of spreadsheets. We first pre-train with the pre-training objectives of local context and sheet structure on the entire dataset for 1M steps. Then we add the objective of analytical semantics and pre-train for another 2.5M steps on 1.54M spreadsheets with formulas. SHEETPT is implemented in PyTorch and uses distributed pre-training on Tesla V100 GPU. The batch size in both two phases is 8.

Downstream Task: Formula Prediction

Formula prediction is a task of high research and practical importance in spreadsheet intelligence. It aims to predict the formula for a given cell. The accurate prediction of formulas requires a powerful understanding of the data semantics.

Dataset. We follow SpreadsheetCoder (Chen et al. 2021) and ForTap(Cheng et al. 2021) to use Enron (Hermans and Murphy-Hill 2015) as the dataset for formula prediction. ForTap has published the dataset of training, validation, and test. Notably, baseline models only handle a table at one time, thus only involving the formulas where reference cells are all inside the table and removing formulas with cross-table reference cells from the datasets. But we expand the datasets by adding back the cross-table formulas and removing extremely large spreadsheets following the pre-training setup. We categorize all the formulas in the validation and test datasets into two types: inside-table formulas (targets of ForTap and SpreadsheetCoder) and cross-table formulas (formulas with reference cells outside the table). The statistics of the augmented datasets are shown in Table 1. Cross-table formulas account for more than 20% of both the test and validation datasets, which is a non-trivial part and can not be ignored.

	Validation	Test
# Sheets	448	474
# Tables per sheet (Avg.)	1.73	1.79
# Formulas	6,669	7,417
# Inside-table formulas	5,187	5,809
# Cross-table formulas	1,482	1,608
% Cross-table formulas	22.2	21.7

Table 1: Statistics of validation and test datasets augmented with cross-table formulas in formula prediction.

Baselines. We compare SHEETPT with two representative baselines, SpreadsheetCoder and ForTaP to verify its effectiveness. SpreadsheetCoder is a dedicated BERT-based model for formula prediction on spreadsheet tables. ForTaP devises formula-related pre-training tasks to enhance numerical reasoning ability for the tree-based transformer model TUTA (Wang et al. 2021) and they both leverage table boundary and structural information (e.g., detected header

cells and their hierarchical relations) in spreadsheets. They are all table-centric models that can only predict inside-table formulas.

Experimental Setup. SHEETPT consumes the entire spreadsheet and is fine-tuned on the formula prediction dataset. In both fine-tuning and evaluation phases, formulas of targeted cells and their neighbor cells from the same chunks are removed from the inputs of SHEETPT to avoid data leakage. For ForTap, we use their released checkpoints fine-tuned on the same training dataset with only inside-table formulas. For SpreadsheetCoder, we leverage the reproduced version by (Cheng et al. 2021). For both baselines, the inputs are created based on explicit metadata of table boundary and structure. Formulas are generated by beam search with a beam size of 5.

Results. We follow the evaluation metrics used in SpreadsheetCoder and ForTap, where results are measured by the accuracy of the top-1 predicted sketch, range and formula. For a more in-depth analysis, we also summarize the results for inside-table formulas and cross-table formulas individually. As shown in Table 2, SHEETPT outperforms baseline models across all the metrics. The overall accuracy of formula prediction is largely improved by more than 18% compared with ForTap. Specifically, SHEETPT is capable of predicting cross-table formulas with an accuracy of 50.19%. While table-centric models can only correctly guess a relatively small part of formula sketches, and fail to predict the reference ranges due to the weakness in cross-table modeling. Besides, for inside-table formulas, SHEETPT also outperforms ForTap by more than 9% even if ForTaP incorporates additional table boundary and structural information, evidencing that SHEETPT implicitly captures the information about table boundary and structure and can use data of the entire sheet to strengthen the understanding of inside-table data.

Models	Formula Type	Formula	Sketch	Range
Spreadsheet-Coder	All	29.60	47.15	62.78
	Inside-table	37.80	58.65	64.46
	Cross-table	0	5.61	0
ForTaP	All	46.80	60.82	76.95
	Inside-table	59.75	74.31	80.40
	Cross-table	0	12.07	0
SHEETPT	All	**64.93**	**78.62**	**82.59**
	Inside-table	**69.01**	**82.17**	**83.99**
	Cross-table	**50.19**	**65.80**	**76.28**

Table 2: Formula prediction accuracy (%) on Enron.

Downstream Task: Sheet Structure Recognition

Understanding sheet structure is critical for many tasks on spreadsheets, and a key step is to identify semantic roles for each cell (Paine 2008; Koci et al. 2016; Gol, Pujara, and Szekely 2019). Performance on this task can well reflect the capability of the model in exploring the semantic structure of sheets. In addition, most of the state-of-the-art approaches for this task are based on table pre-training models and are only applicable to the sheet with a single explicitly marked table boundary. Hence, We conduct experiments to exhibit the performance in recognizing sheet structure.

Dataset. We employ DeEx (Koci et al. 2019a), a widely used dataset for cell type classification of tabular data. Cells in the sheets are categorized into six types: `metadata`, `notes`, `data`, `left attribute`, `top attribute`, and `derived`. Besides, we also annotate a new dataset for sheet structure recognition due to the following limitations of existing public datasets (Koci et al. 2019a; Gol, Pujara, and Szekely 2019; Koci et al. 2016): 1) Existing datasets are relatively small and we have observed great variations when running experiments on these datasets. 2) Most of the spreadsheets in these datasets consist of only a single table while multiple tables in one sheet are also common in practice. 3) The spreadsheet files of these datasets are old and out of date. Therefore, we annotate a new dataset, SHEET-SEM, with 2,801 sheets following the same annotations as DeEx. The spreadsheets are newly crawled from millions of websites and randomly sampled for annotation. Each sheet is labeled and verified by 3 people.

Baselines. We adopt two well-known tabular pre-training models, TaPas (Herzig et al. 2020a) and ForTaP (Cheng et al. 2021) as our baselines. ForTap is the state-of-the-art method for cell type classification of tabular data.

Experimental setup. SHEETPT consumes the entire sheet and provides representations for each cell. Then a simple MLP layer with cross-entropy loss is applied for cell classification. For ForTap and TaPas, as there can be more than one table on a sheet and it is infeasible to feed multiple tables as their inputs, we instead use the spreadsheet boundary as the default table boundary for fairness. We use the released checkpoints and codes of ForTap, and the TaPas model in Hugging Face (Hugging Face Team 2021) for fine-tuning. We conduct a 5-fold cross-validation following the setup of ForTap on the DeEx dataset. As for experiments on SheetSem, we split the dataset into 2,201 training sheets, 300 validation sheets and 300 test sheets. All models with the best Macro F1 on the validation set during fine-tuning are used for the evaluation on the test set.

Results. The F1 scores of each cell type and the Macro average are the evaluation metrics for sheet structure recognition. As Table 3 illustrates, SHEETPT achieves remarkable state-of-the-art results on sheet structure recognition with Macro-F1 of 84.61% on DeEx and 87.95% on SheetSem. Besides, the F1-scores on most of the cell types are raised by SHEETPT compared with baseline models. Specifically, SHEETPT delivers a significant improvement on type `derived` on both datasets with increases of 12.85% and 20.22%. Note that `derived` labels a cell that can be calculated by other cells, and there can be formulas over such cells. Therefore, accurate detection of such cells reveals the great power of SHEETPT in implicitly capturing the semantic relations between cells, especially when there is no formula or cells are distant from each other.

Ablation Studies

To verify the effectiveness of our design, we conduct comprehensive ablation studies on different components of SHEETPT and the pre-training objectives.

Table 4 and Table 5(left) show the ablation results of pre-training objectives on formula prediction and sheet struc-

Cell Types	DeEx			SheetSem		
	TaPas	ForTap	**SheetPT**	TaPas	ForTap	**SheetPT**
M	79.84	76.58	**86.14**	77.36	81.59	**89.68**
N	20.45	53.08	**58.30**	88.54	88.93	**93.72**
Data	99.58	99.27	**99.69**	92.99	92.62	**94.11**
LA	78.61	71.50	**82.63**	**84.81**	83.04	84.23
TA	90.41	83.64	**96.30**	91.66	92.53	**96.40**
Derived	68.61	71.74	**84.59**	48.58	49.33	**69.55**
Avg.	72.92	75.70	**84.61**	80.66	81.44	**87.95**

Table 3: F1-scores (%) of sheet structure recognition on DeEx and SheetSem: M(metadata), N(notes), Data, LA(left attribute), TA(top attribute), and Derived.

Pre-training Objectives	Formula Type	Formula	Sketch	Range
L	All	41.75	61.94	67.40
	Inside-table	44.34	64.31	68.94
	Cross-table	32.40	53.36	60.72
L + S	All	50.11	69.88	71.71
	Inside-table	52.46	72.49	72.38
	Cross-table	41.60	60.45	68.83
L + S + A	All	**64.93**	**78.62**	**82.59**
	Inside-table	**69.01**	**82.17**	**83.99**
	Cross-table	**50.19**	**65.80**	**76.28**

Table 4: Ablation experimental results of pre-training objectives on formula prediction. L: Local context. S: Sheet structure. A: Analytical semantics.

ture recognition. Compared with the variant only pre-trained by the local context objective, adding the sheet structure objective can improve all the evaluation metrics on both downstream tasks. The results inspire us that pre-training with the pseudo labels from table detection models can enhance the structure understanding ability of SHEETPT, and is beneficial to downstream tasks. With an extra pre-training objective of analytical semantics, the performance is further boosted, evidencing that the analytical semantics from spreadsheets data enable SHEETPT to better capture the semantics between cells.

Ablation results of model components on sheet structure recognition are organized in Table 5. Combining all model components (i.e., SHEETPT) achieves the highest Macro-F1 of 87.95%. While removing chunk-level relational attention decreases the Macro-F1 by 0.75%, further removing coarse- or fine-grained attention causes a more evident drop of 1-3%, indicating the importance of chunk attention with multiple granularities for SHEETPT.

Related Work

Table detection and sheet structure recognition Although SHEETPT is the first large-scale pre-training model on spreadsheets, spreadsheets understanding has been widely studied before. Since many previous works (Chen et al. 2021; Wang et al. 2021; Du et al. 2021; Pinto et al. 2003) focus on spreadsheet tables and rely on explicit boundary and structure information (e.g., headers), table detection and structure recognition serve as a fundamental technique for table-centric approaches. (Dong et al. 2019b) leverages CNN (He et al. 2016) to detect table, (Chen and Cafarella 2013) uses heuristic methods to extract relational information from data, (Koci et al. 2019b) adopts graph model to identify the tabular payload from various metadata, and (Dong et al. 2019a) utilizes multi-task learning to jointly learn the table region and structural information. Our methods explore using the pre-training technique to enhance spreadsheets understanding.

Cell Types	Objectives			Models			
	L	L+S	L+S+A	I	II	I+II	I+II+III
M	74.73	83.17	**89.68**	85.91	88.01	86.27	**89.68**
N	81.80	93.68	**93.72**	91.64	91.64	**95.21**	93.72
Data	91.83	93.39	**94.11**	92.96	93.85	94.05	**94.11**
LA	77.60	80.19	**84.23**	82.83	82.14	**85.28**	84.23
TA	92.56	93.31	**96.40**	94.55	**96.49**	95.24	96.40
Derived	59.65	67.69	**69.55**	60.28	68.90	67.13	**69.55**
Avg.	79.69	85.24	**87.95**	84.62	86.84	87.20	**87.95**

Table 5: Ablation experimental results of sheet structure recognition on SheetSem. L: Local context. S: Sheet structure. A: Analytical semantics. I: Coarse-grained attention. II: Fine-grained attention. III: Chunk-level relational attention.

Table pre-training Table pre-training has been well studied in recent years. Some works (Herzig et al. 2020b; Yin et al. 2020; Liu et al. 2021) target table understanding of relational tables, and some (Wang et al. 2021) expand the scope for structured tables. Simultaneously, a range of various applications on tables are widely explored with pre-training methods, such as semantic parsing (Herzig et al. 2020b), question answering (Yin et al. 2020; Wang et al. 2021), cell type classification(Wang et al. 2021; Cheng et al. 2021) and formula prediction (Cheng et al. 2021). SHEETPT refers to the design of table pre-training approaches and refining the model architecture and pre-training tasks with regard to the particularity of spreadsheets.

Conclusion

In this paper, we present SHEETPT, a novel pre-training model based on hierarchical attention network for spreadsheet understanding. SHEETPT is the first large-scale pre-training model designed for spreadsheets and enables various downstream tasks on spreadsheets. We propose coherent chunk, an intermediate data granularity in spreadsheets, and elaborate two critical components to capture the contextual information in spreadsheets: hierarchical attention with multiple granularities and chunk-level relational attention based on formulas. Besides, we devise three kinds of pre-training objectives to enhance representation learning at token, cell, and chunk levels. Experiments show that SHEETPT can achieve state-of-the-art performance on downstream formula prediction and sheet structure recognition tasks.

SHEETPT is capable of empowering other downstream tasks in spreadsheets, such as table detection and chart recommendation. Exploiting the potential of SHEETPT for various spreadsheet applications will be a promising research direction in the near future.

References

Chen, X.; Maniatis, P.; Singh, R.; Sutton, C.; Dai, H.; Lin, M.; and Zhou, D. 2021. Spreadsheetcoder: Formula prediction from semi-structured context. In *International Conference on Machine Learning*, 1661–1672. PMLR.

Chen, Z.; and Cafarella, M. 2013. Automatic web spreadsheet data extraction. In *Proceedings of the 3rd International Workshop on Semantic Search over the Web*, 1–8.

Cheng, Z.; Dong, H.; Cheng, F.; Jia, R.; Wu, P.; Han, S.; and Zhang, D. 2021. FORTAP: Using Formulae for Numerical-Reasoning-Aware Table Pretraining. *arXiv preprint arXiv:2109.07323*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, H.; Cheng, Z.; He, X.; Zhou, M.; Zhou, A.; Zhou, F.; Liu, A.; Han, S.; and Zhang, D. 2022. Table Pretraining: A Survey on Model Architectures, Pretraining Objectives, and Downstream Tasks. *arXiv preprint arXiv:2201.09745*.

Dong, H.; Liu, S.; Fu, Z.; Han, S.; and Zhang, D. 2019a. Semantic structure extraction for spreadsheet tables with a multi-task learning architecture. In *Workshop on Document Intelligence at NeurIPS 2019*.

Dong, H.; Liu, S.; Han, S.; Fu, Z.; and Zhang, D. 2019b. Tablesense: Spreadsheet table detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 69–76.

Du, L.; Gao, F.; Chen, X.; Jia, R.; Wang, J.; Zhang, J.; Han, S.; and Zhang, D. 2021. TabularNet: A neural network architecture for understanding semantic structures of tabular data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 322–331.

Gol, M. G.; Pujara, J.; and Szekely, P. 2019. Tabular cell classification using pre-trained cell embeddings. In *2019 IEEE International Conference on Data Mining (ICDM)*, 230–239. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hermans, F.; and Murphy-Hill, E. 2015. Enron's spreadsheets and related emails: A dataset and analysis. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, 7–16. IEEE.

Herzig, J.; Nowak, P. K.; Müller, T.; Piccinno, F.; and Eisenschlos, J. M. 2020a. TaPas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Herzig, J.; Nowak, P. K.; Müller, T.; Piccinno, F.; and Eisenschlos, J. M. 2020b. TaPas: Weakly Supervised Table Parsing via Pre-training. *ArXiv*, abs/2004.02349.

Huang, Z.; and He, Y. 2018. Auto-detect: Data-driven error detection in tables. In *Proceedings of the 2018 International Conference on Management of Data*, 1377–1392.

Hugging Face Team. 2021. TAPAS model in Hugging Face. https://huggingface.co/transformers/v4.8.0/model_doc/tapas.html. Accessed: 2023-03-07.

Koci, E.; Thiele, M.; Rehak, J.; Romero, O.; and Lehner, W. 2019a. DECO: A dataset of annotated spreadsheets for layout and table recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1280–1285. IEEE.

Koci, E.; Thiele, M.; Romero, O.; and Lehner, W. 2019b. A Genetic-Based Search for Adaptive Table Recognition in Spreadsheets. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1274–1279.

Koci, E.; Thiele, M.; Romero Moral, Ó.; and Lehner, W. 2016. A machine learning approach for layout inference in spreadsheets. In *IC3K 2016: Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management: volume 1: KDIR*, 77–88. SciTePress.

Liu, Q.; Chen, B.; Guo, J.; Lin, Z.; and Lou, J.-g. 2021. Tapex: Table pre-training via learning a neural sql executor. *arXiv preprint arXiv:2107.07653*.

Microsoft. 2022a. AGGREGATE function. https://support.microsoft.com/en-us/office/aggregate-function-43b9278e-6aa7-4f17-92b6-e19993fa26df. Accessed: 2023-03-07.

Microsoft. 2022b. Number format codes. https://support.microsoft.com/en-us/office/number-format-codes-5026bbd6-04bc-48cd-bf33-80f18b4eae68. Accessed: 2023-03-07.

Paine, J. 2008. Spreadsheet structure discovery with logic programming. *arXiv preprint arXiv:0802.3940*.

Pinto, D.; McCallum, A.; Wei, X.; and Croft, W. B. 2003. Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 235–242.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *NAACL*.

Wang, Z.; Dong, H.; Jia, R.; Li, J.; Fu, Z.; Han, S.; and Zhang, D. 2021. TUTA: tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1780–1790.

Yin, P.; Neubig, G.; Yih, W.-t.; and Riedel, S. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.

Zhang, Y.; Lv, X.; Dong, H.; Dou, W.; Han, S.; Zhang, D.; Wei, J.; and Ye, D. 2021. Semantic table structure identification in spreadsheets. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 283–295.

Zhou, M.; Li, Q.; He, X.; Li, Y.; Liu, Y.; Ji, W.; Han, S.; Chen, Y.; Jiang, D.; and Zhang, D. 2021. Table2Charts: Recommending Charts by Learning Shared Table Representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2389–2399.