**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; Cheng 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 85th 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
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 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
In spreadsheets, there is rich information in cell ranges as a result of various formatting 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 adjacent 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 and fine-grained levels, are devised to extract thorough semantics efficiently.

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, and 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.

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:

$$E_{CELL} = E_{SMT} + E_{POS} + E_{NUM} + E_{FORMAT}.$$  (1)
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 H5:115 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 $R \in \mathbb{R}^{d_{\text{Rel}}}$ with $d_{\text{Rel}}$ dimension. Similar to relative positional embedding (Shaw, Uszkoreit, and Vaswani 2018), Chunk-Level Relational Attention (CLRA) can be written as

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

**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 $L_{\text{MLM}}$ and $L_{\text{CLC}}$ 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 $L_{\text{SRC}}$.

**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 $L_{\text{NFTP}}$ 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 $L_RP$.

The overall pre-training objective can be formulated as

$$L = L_{\text{MLM}} + L_{\text{CLC}} + L_{\text{SRC}} + L_{\text{NFTP}} + L_{\text{RP}}. \quad (3)$$
Table 1: Statistics of validation and test datasets augmented with cross-table formulas in formula prediction.
	est. 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.

**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 re-produced 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.

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

<table>
<thead>
<tr>
<th>Models</th>
<th>Formula Type</th>
<th>Formula</th>
<th>Sketch</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spreadsheet-Coder</td>
<td>All</td>
<td>29.60</td>
<td>47.15</td>
<td>62.78</td>
</tr>
<tr>
<td></td>
<td>Inside-table</td>
<td>37.80</td>
<td>58.65</td>
<td>64.46</td>
</tr>
<tr>
<td></td>
<td>Cross-table</td>
<td>46.80</td>
<td>60.82</td>
<td>76.95</td>
</tr>
<tr>
<td>ForTap</td>
<td>All</td>
<td>59.75</td>
<td>74.31</td>
<td>80.40</td>
</tr>
<tr>
<td></td>
<td>Inside-table</td>
<td>0</td>
<td>5.61</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Cross-table</td>
<td>64.93</td>
<td>78.62</td>
<td>82.59</td>
</tr>
<tr>
<td>SHEETPT</td>
<td>All</td>
<td>69.01</td>
<td>82.17</td>
<td>83.99</td>
</tr>
<tr>
<td></td>
<td>Inside-table</td>
<td>50.19</td>
<td>65.80</td>
<td>76.28</td>
</tr>
</tbody>
</table>

**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 SheetSemi, 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 SheetSemi. 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-
DeEx and SheetSem: M(metadata), N(notes), Data, LA(left attribute), TA(top attribute), and Derived.

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.

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


Related Work

Table detection and sheet structure recognition Although 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.

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.
References


