Check for updates

RESEARCH PAPER

HUSS: A Heuristic Method for Understanding the Semantic Structure of Spreadsheets

Xindong Wu[1,2†], Hao Chen[1], Chenyang Bu[1], Shengwei Ji[1], Zan Zhang[1], Victor S. Sheng[3]

[1]Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, China, School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China

[2]Research Institute of Artificial Intelligence, Zhejiang Lab, Hangzhou, China

[3]Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

Keywords: Spreadsheet semantic structure; Information extraction; Heuristics; Cell function analysis; Table structure analysis

Citation: Wu, X.D., Chen, H., Bu, C.Y., et al.: HUSS: A heuristic method for understanding the semantic structure of spreadsheets. Data Intelligence 5(3), 537-559 (2023). doi: 10.1162/dint_a_00201
Submitted: November 30, 2022; Revised: December 16, 2022; Accepted: January 16, 2023

ABSTRACT

Spreadsheets contain a lot of valuable data and have many practical applications. The key technology of these practical applications is how to make machines understand the semantic structure of spreadsheets, e.g., identifying cell function types and discovering relationships between cell pairs. Most existing methods for understanding the semantic structure of spreadsheets do not make use of the semantic information of cells. A few studies do, but they ignore the layout structure information of spreadsheets, which affects the performance of cell function classification and the discovery of different relationship types of cell pairs. In this paper, we propose a Heuristic algorithm for Understanding the Semantic Structure of spreadsheets (HUSS). Specifically, for improving the cell function classification, we propose an error correction mechanism (ECM) based on an existing cell function classification model [11] and the layout features of spreadsheets. For improving the table structure analysis, we propose five types of heuristic rules to extract four different types of cell pairs, based on the cell style and spatial location information. Our experimental results on five real-world datasets demonstrate that HUSS can effectively understand the semantic structure of spreadsheets and outperforms corresponding baselines.

† Corresponding author: Xindong Wu (E-mail: xwu@hfut.edu.cn; ORCID: 0000-0003-2396-1704).

© 2023 Chinese Academy of Sciences. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

1. INTRODUCTION

Spreadsheet is a computer application for computation, organization, analysis and storage of data in a tabular form. With the development of the information age, spreadsheets are widely used in different fields, such as government, business, and scientific research [1, 2, 3]. Moreover, spreadsheet data also has many practical applications, such as table search [4, 5], and query answering [6]. Understanding the semantic structure of spreadsheets, i.e., table understanding, is a key technique for these practical applications.

Understanding the semantic structure of spreadsheets includes two key steps, (i) cell function analysis and (ii) table structure analysis [7]. First, spreadsheets consist of a large number of cells that play different functional types, such as header, data, attributes, as shown in Figure 1 (a). Therefore, understanding a spreadsheet needs to analyze the functional types of cells in a spreadsheet, assigning a predefined functional label to each cell in the spreadsheet. Second, spreadsheets are designed for human comprehension [8]. The preference of different people leads to spreadsheets with flexible and diverse layout formats, which results in spreadsheet data usually organized in a semi-structured form. In addition, in order to make spreadsheet data more effective for users to read and compare, different areas of a real-world spreadsheet use different ways to display data, and thus have different layout characteristics [9]. Therefore, understanding a spreadsheet needs to analyze its layout structure, discovering cell pairs of different relationship types in the spreadsheet to resolve hierarchical index relationships within different types of areas.

For example, the header area of the spreadsheet uses features such as merged cells and hierarchies to display data, the attribute area is composed of several sub-areas divided by flag cells, and indented cells are used inside the sub-areas to display data, and the cells of each data area are described by cells in its header area and attribute area. The layout features in different areas of the spreadsheet result in different types of correspondence between cells. As shown in Figure 1 (b), header cells [D1:E1] and [E2] are parent-child relationships, and as shown in Figure 1 (c), attribute cells [A5] and [A6] are sibling relationships. In addition, the data area cells in the spreadsheet are described by the header area and the attribute area cells. As shown in Figure 1 (a), the data cell [C5] and the header cell [C2] are the header index relationship, the data cell [C5] and the attribute cell [A5] are the attribute index relationship. Based on the table understanding solution proposed by Hurst [10], understanding the semantic structure of spreadsheets can be solved through (i) **cell function analysis** and (ii) **table structure analysis**.

Existing methods for understanding the semantic structure of spreadsheets can be categoried into these two groups (i.e., the cell function analysis [11, 12, 13, 14] and the cell function and table structure analysis [15, 16, 17, 18, 19, 20, 21]. Among the methods for cell function analysis, Chen et al. [12], Koci et al. [13], and Adelfio et al. [14] use the manual styling, formatting, and typography features of cells, but these features are usually found only in well-formatted documents. Therefore, they are not universal. Gol et al. [11] proposed an RNN based on cell context, semantic information vector representation, and their previous cell style information to classify the function type of each cell in complex spreadsheets. Chen et al. [15] [16, 17] and Shigarov et al. [18, 19, 20] use heuristic rules and cell style characteristic information to classify cells. They ignore cell context and semantic information, resulting in a poor cell classification accuracy especially when spreadsheets are with different fields.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

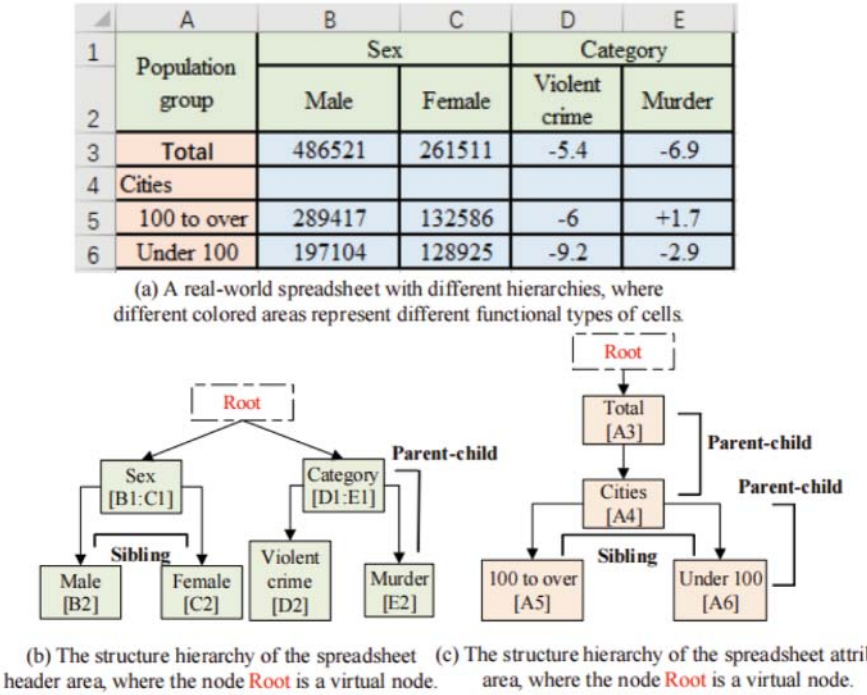

Figure 1. The layout features in different areas of the spreadsheet. (a) A real-world spreadsheet with a complex hierarchy and different cell function types. (b) The hierarchies in the header area at the top of the spreadsheet. (c) The attribute area at the left of the spreadsheet.

Among the methods for the table structure analysis, Chen et al. [15, 16, 17] only extracted parent-child cell pairs. Shigarov et al. [18, 19, 20] can only deal with spreadsheets in specific fields, and do not make use of the layout characteristics of attribute regions in spreadsheets. Pujara et al. [21] proposed a table understanding paradigm, including cell classification, block detection, and layout prediction. But it focuses on the data type classification of cells rather than the functional types of cells. It discovers the relationships between blocks in the table (an area of cells with the same data type), instead of the table structure.

Based on the above problems, we propose a combined heuristic spreadsheet semantic structure understanding algorithm (HUSS) to analyze cell function and table structure simultenously. Specifically, we propose an Error Correction Mechanism (ECM) to effectively improve the cell function type classification accuracy by using the layout features of spreadsheets. Besides, we also propose five types of heuristic rules to extract four relation types of cell pairs in the table structure analysis, based on the functional type, style, and spatial location characteristics of the cells. This research is an extension of our earlier accepted paper in 2022 IEEE International Conference on Knowledge Graph (ICKG) [29]. The main contributions of this paper are summarized as follows.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

- A combined heuristic spreadsheet semantic structure method (HUSS) is proposed, which can effectively identify the functional types of cells and extract cell pairs of four relation types.
- Based on the layout features of spreadsheets, an error correction mechanism (ECM) is proposed to further improve the cell function classification accuracy.
- Based on the functional types, styles, and spatial location characteristics of cells, we propose five types of heuristic rules to effectively discover four different types of cell pairs in spreadsheets.
- Experimental results on five real-world spreadsheet datasets demonstrate that HUSS can effectively identify cell types and discover four different types of cell pairs.

This remainder of this paper is structured as follows. Section 2 introduces the related work of this study. Section 3 introduces preliminaries, including terms and definitions for the cell function analysis and the table structure analysis. Our method HUSS is explained in detail in Section 4. Our experimental results on five real datasets are shown in Section 5. Finally, Section 6 concludes the paper.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

2. RELATED WORK

The problem of understanding spreadsheets (i.e., extracting valuable patterns and data from spreadsheets and converting them into a machine-readable structure form) has been extensively studied. According to the table understanding methods proposed by Hurst [10], an end-to-end solution to generate machine-readable structured information from spreadsheets should consist of following five parts (i.e., localization, segmentation, interpretation, cell functional analysis and table structure analysis).

Although the cell function analysis and the table structure analysis are divided into two separate parts by Hurst [10], the cell function analysis is usually the basis of the table structure analysis. The purpose of the cell function analysis is to determine the function type of cells, while the structure analysis is to find the relationship types of cell pairs. Therefore, we present the cell function analysis and the table structure analysis together here.

The exist many research articles on the cell function analysis and/or the table structure analysis [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. Chen et al. [12], Koci et al. [18], and Adelfio et al. [14] focus on solving cell function analysis by classification utilizing cell styles, formast, and typographical characteristics, such as cell background colors, font sizes, data types, etc. However, these features are usually found only in richly formatted documents, so these cell function classification methods are not universally applicable.

Chen et al. [15, 16, 17] focuses on automatic extraction of relational data from spreadsheets. For their cell function analysis, they use a Conditional Random Field (CRF) classifier suitable for sequential classification to predict the type of the current row by using the characteristics of the previous row. Finally, each row of the spreadsheet is assigned a predefined label, such as Title, Header, Data, and Footnote. But CRF also relies on the styles, formatting, and typographical characteristics of cells, and is not effective when dealing with documents with complex layout structures. For the table structure analysis, some machine learning techniques (e.g., SVM and EN-SVM) are used to extract only parent-child cell pairs. Shigarov

et al. [18, 19, 20, 27] focus on converting semi-structured spreadsheet data into a relational form. They use heuristic rules to solve the cell function classification and the table structure analysis, and a domain-specific language (Cells Rule Language) is proposed to execute these predefined rules. However, these approaches only work well with domain specific spreadsheets.

Recently, an RNN classification technique based on cell context and semantic information vector representation is proposed by [11], which combines with previous cell style information to improve the performance of cell function classification in complex layout structure documents. However, [11] only focuses on cell function analysis, and is not for the table structure analysis. Based on the table understanding paradigm proposed in [21], Sun et al. [8] divides table understanding into three parts: cell function classification, block detection, and layout prediction. However, its cell function classification focuses on the data type classification of cells rather than their function types. Besides, its table structure analysis is to discover the relationships between blocks in spreadsheets (a block is an area of a table with the same type of cells). However, we focus on discovering relationships between cells to understand the table structure.

3. PRELIMINARIES

Before introducing our approach, the preliminaries of understanding the semantics structure of spreadsheets will be introduced as follows. We first provide related problem and term definitions, and then provide notations and corresponding descriptions used in the paper.

3.1 Problem and Related Term Definition

As introduced in Section I, the problem of understanding the semantic structure of spreadsheets includes cell function analysis and the table structure analysis. The cell function analysis is to classify cells according to pre-defined cell types. The table structure analysis is to discover pairs of cells with different relationship types.

Existing methods [11, 13, 16, 26] have different definitions and terms for the layout of cells in the spreadsheet. We summarize the terms and their corresponding definitions in the literature in terms of cell function types, which are defined as follows.

Definition 1. Header (*H*). The header is the column head of a table and can usually be hierarchical.

Definition 2. Attribute (*A*). The attribute is the table row header, and like a header, it can be hierarchical.

Definition 3. Data (*D*). The data cell is the core body of a table.

The definitions and terms for the relationship types between cells and blocks in spreadsheets [8, 9] are also different. We summarize these definitions and terms in the literature and define four main relationship types for a given cell pair $< c1, c2 >$. Note that in a cell pair $< c1, c2 >$, we require that $c1$ is to the left of $c2$, otherwise at the top of $c2$. The four specific cell pair relationship types are defined as follows.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

Definition 4. Parent-child relationship type (R_1). The header and attribute areas in a spreadsheet typically contain hierarchies. if $c1$ and $c2$ are header cells, and $c1$ is the parent of h2 in the hierarchy, then the relationship of the cell pairs < $c1$, $c2$ > is parent-child.

Definition 5. Sibling relationship type (R_2). In the header and attribute areas of the spreadsheet, if cells $c1$ and $c2$ have the same parent cell, the relationship of the cell pair < $c1$, $c2$ > is sibling.

Definition 6. Header index relationship type (R_3). In a spreadsheet, each data cell is typically described by a header cell.

Definition 7. Attribute index relationship type (R_4). Similar to $R3$, in a spreadsheet, each data cell is typically described by an attribute cell.

As shown in Figure 1 (b), the cell pair < Category, Violent crime > and < Category, Murder > are both of type R_1. The cell pair < Male, Female > and < Violent crime, Murder > are of type R_2. As shown in Figure 1 (a), the cell pairs < Male, 289417 > and < 100 to over, 289417 > are of type R_3 and R_4, respectively.

3.2 Notations

Here we list the notations and their corresponding descriptions used since then, which is shown in Table 1.

Table 1. Notations and descriptions used in the paper.

Notations	Descriptions
C	A cell
D	Data type cell
H	Header type cell
A	Attribute type cell
R_1	Parent-child type cell pairs
R_2	Sibling type cell pairs
R_3	H index type cell pairs
R_4	A index type cell pairs
H_{set}	The initial set of H cells
D_{set}	The initial set of D cells
A_{set}	The initial set of A cells
$R1_{set}$	R_1 type cell pair set
$R2_{set}$	R_2 type cell pair set
$R3_{set}$	R_3 type cell pair set
$R4_{set}$	R_4 type cell pair set
Col_C	Column coordinates of cell C
Row_C	Row coordinates of cell C
Type_C	The functional type of cell C
Width_C	The column span of cell C
Length_C	The row span of Cell C
AttrNum_C	The block number of the attribute sub-region where cell C is located
IndentNum_C	The number of indents for cell C

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

4. HEURISTIC ALGORITHMS FOR UNDERSTANDING THE SEMANTIC STRUCTURE OF SPREADSHEETS

As we mentioned before, existing methods for understanding the semantic structure of spreadsheets can be categoried into these two groups (i.e., the cell function analysis [11, 12, 13, 14] and the cell function and table structure analysis [15, 16, 17, 18, 19, 20, 21]. Existing studies for the cell function analysis use cell styles, formatting, and typographical features to construct classifiers. They ignore the semantic information of cells, resulting in a poor cell function classification accuracy in spreadsheets with cross-domain and different layout structures. Due to limitations in the cell function analysis, most existing methods for the table structure analysis are also limited to domain-specific spreadsheets. Some studies, such as [11], make full use of the semantic information of cells, but they ignore the information of layout characteristics of the attribute area of spreadsheets. These motivate us first to design heuristic algorithms for understanding the semantic structure of spreadsheets (HUSS). The overview of HUSS is described as follows.

4.1 Overview

The goal of our proposed method HUSS is to transform a semi-structured spreadsheet into a structured form that can be understood by machines, as shown in Figure 2. Specifically, HUSS takes a spreadsheet as input, classifies the function types of each cell in the spreadsheet, and then extracts the cell pairs relationships based on the results of cell function type classification. Classifying the cell function types is based on the cell function classification model proposed in [11]. To further improve its performance, we propose an error correction mechanism (ECM) to detect and relabel misclassified cells by the cell classification model. With the results of cell function type classification, cell styles, and cell spatial location characteristics, we propose five types of heuristic rules to extract cell pairs with different relationship types. Finally, the extracted table information is converted into relational form and stored in an external database. The general description of HUSS is shown in Algorithm 1.

Algorithm 1. The general framework of HUSS.

1. **Input**: D_{set}, H_{set}, A_{set}
2. **Output**: $R1_{set}$, $R2_{set}$, $R3_{set}$, $R4_{set}$
3. $R1_{set}$, $R2_{set}$, $R3_{set}$, $R4_{set} \leftarrow \{\}$;
4. **for** a_i, d_i, h_i in A_{set}, D_{set}, H_{set} **do**
5. Assign an H and A cell to each D cell, according to Algorithm 2 in Section 4;
6. A_{set} is divided into several sub-region sets segmented by C_{flag} cell, according to Algorithm 4 in Section 4;
7. Find cell pairs of type R_1 and R_2 in the attribute area of the table, according to Algorithm 5 in Section 4;
8. **end for**
9. **return** $R1_{set}$, $R2_{set}$, $R3_{set}$, $R4_{set}$;

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

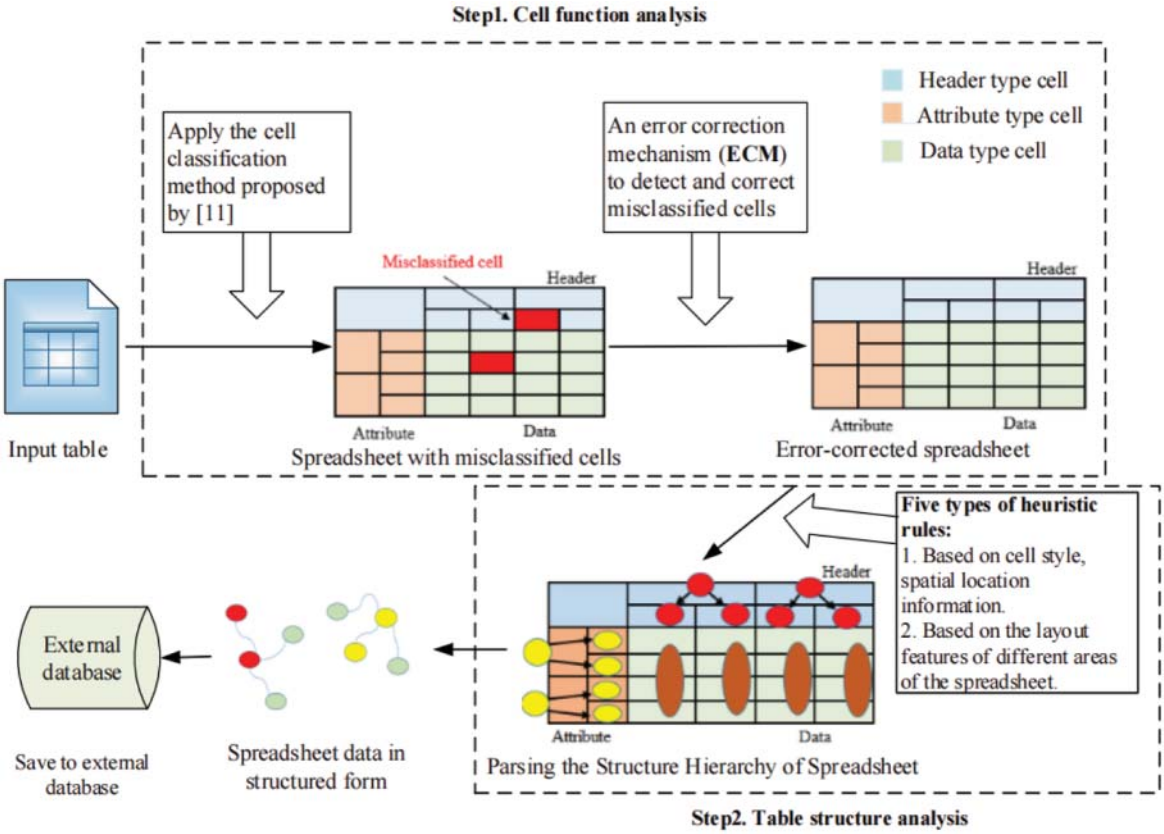

Figure 2. Our proposed method HUSS consists of two components for the cell function analysis and the table structure analysis. The cell function analysis is carried out based on the classification model of cells proposed in [11] with our proposed error correction mechanism (ECM). The table structure analysis utilizes our proposed five types of heuristic rules to extract cell pairs of four relationship types based on the cell function types, styles, spatial location, and layout characteristics of the attribute area in the spreadsheet.

4.2 Cell Functional Analysis

As we mentioned before, existing methods for understanding the semantic structure of spreadsheets ignore the semantic information of cells and the layout features of spreadsheets in the cell function analysis. And the cell classification model proposed in [11] makes full use of the semantic information of cells, but ignores the information of layout characteristics of the attribute area of spreadsheets. Therefore, we propose an error correction mechanism (ECM) for detecting and relabel misclassified cells using the layout features of spreadsheets to further improve the cell function classification accuracy. In all, the cell function analysis in HUSS consists of the following two parts: cell function classification and error correction mechanism.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

- **Cell function classification.** The cell function classification method proposed by [11] uses neural networks to learn the embedding of cells (i.e., the vector representation of table cells). The embedding vector of each cell is composed of two parts. First, the author considers that cells within the same row in the table usually have certain semantic connection, so the content semantic information of a cell is captured by contextual embedding. Secondly, given the rich formatting, style, and typeset features of cells in the table, stylistic embedding is used to encapsulate the characteristic information of cells. There are different ways to capture information about cell context and style characteristics. The author uses a pretrained language model to identify the local context information of cells and uses an automatic coding machine to encode the stylistic feature information of cells. HUSS selects the cell function classification model and pretrains the cell function classification model learned from a large number of tables.
- **Error correction mechanism.** Since HUSS is based on the cell function classification [11], some cell function types could be misclassified. To further improve the cell function classification performance, HUSS detects the cells that are misclassified and reassigns them to other cell types, using our proposed ECM. ECM is based on the assumption that spreadsheets are usually composed of three rectangular regions with different types (i.e., header, attribute, data). First, for a spreadsheet, ECM will use heuristic rules to find rectangular areas to determine the scope of the three functional type areas. Next, the ECM will find the cells that are incorrectly classified, that is, cells that are different from the type of the area. Finally, the ECM will modify the type of these misclassified cells to the type of the region they are in.

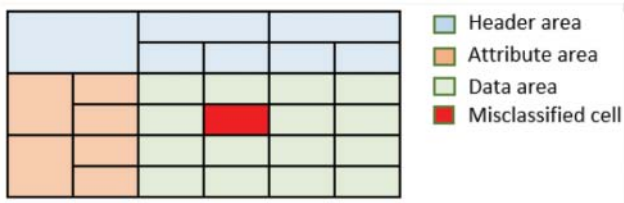

Figure 3. The general layout features of the spreadsheet: The header area is at the top, the attribute area is on the left, and the rest is the data area. Different colors represent spreadsheet areas of different feature types, where red cells are misclassified cells.

4.3 Table Structural Analysis

As we mentioned before, existing methods for understanding the semantic structure of spreadsheets rely extremely on the accuracy of the cell function classification and ignore the layout feature information of the attribute region of the spreadsheet. To improve the table structural analysis, we propose five types of heuristic rules to discover cell pairs in spreadsheets with four different relationship types using cell functional types, styles, spatial feature information, and layout feature information of the attribute region. It contains three major steps: (i) Match data (D) cell, (ii) Match header (H) cell, and (iii) Match attribute (A) cell. In the following, we first introduce the proposed five types of heuristic rules, and then describe in detail the three steps for extracting the cell pairs of the four relationship types.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

Rule-Set. We propose five types of heuristic rules to help parse the structure of the spreadsheet. Rules I and II are used for the Match D cell step, Rule III is used for the Match H cell step, and Rules IV and V are used for the Match A cell step. The rules are as follows.

- **Rule I.** Given a cell pair (c1, c2), if Type_c1 is H or A, Type_c2 is D, and satisfy either condition one (Row_c1 = Ro2_c2 and Width_c1 = Width_c2) or condition two (Col_c1 = Col_c2 and Length_c1 = Length_c2), then the relationship type of the cell pair (c1, c2) is R_3 or R_4.
- **Rule II.** Given a cell pair (c1, c2), if both Type_c1 and Type_c2 are D, and satisfy either of the conditions Row_c1 = Row_c2, or Col_c1 = Col_c2, then cells c1 and c2 correspond to the same H and A cells.
- **Rule III.** Given a cell pair (c1, c2), if both Type_c1 and Type_c2 are H, and satisfy either condition one (Row_c1 = Row_c2 and Wid_c1 = Wid_c2) or condition two (Col_c1 = Col_c2 and Length_c1 = Length_c2), then the relationship type of the cell pair (c1, c2) is R_1.
- **Rule IV.** Given a cell c, if Type_c is A, and c has the most style characteristics (such as bold, centered), then c is the flag cell C_{flag}.
- **Rule V.** Given a cell pair (c1, c2), if both Type_c1 and Type_c2 are A, and satisfy the conditions Col_c1 = Col_c2, IndentNum_c1 = IndentNum_c2, and AttrNum_c1 = AttrNum_c2, then the relationship type of the cell pair (c1, c2) is R_1.

Match data D cell. As introduced in Section I, each D cell in the data area of the spreadsheet is described by an H cell in the header area and A cell in the attribute area. For example, as shown in Figure 4, the D cell (2) is described by the H cell (h3) and the A cell (A1). This step is dedicated to discovering cell pairs of type R_3 and R_4 by assigning an H and A cell to each D cell using the cell's spatial location information. Suppose $D_{set} = \{d_1, d_2, d_3, \cdots\}$, $H_{set} = \{h_1, h_2, h_3, \cdots\}$, and $A_{set} = \{a_1, a_2, a_3, \cdots\}$ are the sets of cells in the set represents a cell, and in order to effectively utilize the spatial relationship between cells, each cell is saved in the form of "(a, b) : (c, d) : text", where a, b represent the row and column coordinates of the upper left corner of the cell, c, d represent the row and column coordinates of the lower right corner of the cell, and text is the content of the cell. As shown in Figure 4, cell AB is saved as "(2, 2):(3, 4) : AB". The algorithm first traverses sets D_{set}, H_{set}, A_{set}, if the cell pair (h_i, d_j) or (a_i, d_j) satisfies Rule I, then it will be saved to $R3_{set}$ or $R4_{set}$, respectively. After the above steps are completed, D_{set} is divided into two sets, i.e., the matched D cell set $D_{matched}$ and the unmatched D cell set $D_{un\_matched}$, as shown in Figure 4. Next, traverse the set $D_{un\_matched}$, if the cell pair (d_i, d_j) satisfies Rule II, then the header and attribute cells corresponding to d_i and d_j are the same, and the corresponding cell pairs are saved to $R3_{set}$ and $R4_{set}$, respectively. The process of matching data cells is shown in Algorithm 2.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

Algorithm 2. Match data cells.

```
1. Input: D_set, H_set, A_set
2. Output: R3_set, R4_set
3. for a_i, d_i, h_i in A_set, D_set, H_set do
4.     if (h_i, d_i), (a_i, d_i) satisfy Rule I then
5.         R3_set.add(h_i, d_i);
6.         R4_set.add(a_i, d_i);
7.     end if
8. end for
9. for d_i, d_j in D_set do
10.    if (d_i, d_j) satisfy Rule II then
11.        R3_set.add(h_i, d_i);
12.        R4_set.add(a_i, d_i);
13.    end if
14. end for
15. return R3_set, R4_set;
```

Match header H cell. As introduced in Section I, the spreadsheet header area usually uses a hierarchical structure to display data. This step is dedicated to parsing the hierarchy within the header area of the spreadsheet to discover all header cell pairs of type R_1 and R_2. After matching the data cells in the previous step, all cell pairs of type R_3 and R_4 are found. At the same time, the set H_{set} is divided into a set $H_{matched}$ of matched data cells and a set $H_{un\_matched}$ of unmatched data cells. The algorithm first traverses the sets $H_{matched}$ and $H_{un\_matched}$. If the cell pair ($d_{matched\_i}$, $d_{unmatched\_j}$) satisfies Rule III, they are stored in the set $R1_{set}$, and all header cell pairs with the same parent cell are stored into the collection $R2_{set}$. The process of matching header cells is shown in Algorithm 3.

Algorithm 3. Match header cells.

```
1. Input: H_set
2. Output: R1_set, R2_set
3. for h_i, h_j in H_set do
4.     if (h_i, h_j) satisfy Rule III then
5.         R1_set.add(h_i, h_j);
6.     end if
7. end for
8. for (h_a, h_b), (h_c, h_d) in R1_set do
9.     if h_b == h_d then
10.        R2_set.add(h_a, h_c);
11.    end if
12. end for
13. return R1_set, R2_set;
```

Match attribute A cell. This step is devoted to find A cell pairs of type R_1 and R_2 in the attribute area of the spreadsheet. Cells in the attribute area of a spreadsheet often contain different types of styling features, such as the number of indents, bolding, centering, etc. In general, these various cell style features provide clues for resolving cell hierarchies. From the above description, we propose a divide-and-conquer

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

algorithm for parsing the cell hierarchy of the spreadsheet attribute area. The general steps of the divide-and-conquer algorithm include: *decomposition*, dividing the problem to be solved into several smaller-scale similar problems; *solving*, when the sub-problems are sufficiently small, solve them by appropriate methods; *merging*, according to the requirements of the original problem, combine the solutions of the sub-problems to form the solution of the original problem). Specifically, this algorithm includes following three sub-steps.

- **Dividing into sub-regions.** The purpose of this step is to divide the original attribute area into a set of several sub-areas $A_{sub\_set} = \{attr_1, attr_1, attr_3, \cdots\}$. Each sub-area is divided by a flag cell C_{flag} with the same style characteristics (such as bold, centered), as shown in Figure 4. The algorithm first traverses the attribute area. When the cell a_i satisfies Rule IV, a_i is marked as the flag cell C_{flag}, and all subsequent cells with the same style characteristics as a_i are recorded as C_{flag} until the end of the traversal. The algorithm description is shown in Algorithm 4.
- **Solution of sub-regions.** For the sub-region set $Attr_{set}$ obtained in the previous step, the algorithm traverses each sub-region $Attr_i$. The number of indents cells contained within a sub-region is usually an indication of the hierarchical relationships between cells. For example, the indents of cells A, A1, A11, A12, and B in Figure 4 are 0, 2, 4, 4, 0 respectively, so that the relationship types of cell pairs (A, A1), (A1, A11), (A1, A12), and (A, B) can be deduced to be R_1, R_1, R_1, R_2, and R_2 respectively. Although cells A1, B1, and C1 are all indented by 2, there is no relationship between cell pairs (A1, B1) and (A1, C1). The former is because cells A1 and B1 belong to different sub-regions. Therefore, traverse the cells in $Attr_i$, and store the cell pair (a_i, a_j) satisfying Rule V into $R1_{set}$. The cell pair (a_i, a_k) with the same parent cell is stored in $R2_{set}$. Repeat the above process for all sub-regions $Attr_i$ can be found. The algorithm process is shown in Algorithm 5.
- **Merge.** The solution set of each sub-region $Attr_i$ obtained in the previous step contains all R_1 and R_2 cell pairs in the corresponding sub-regions. Then we form parent-child cell pairs with the non-parent cell in each sub-region $Attr_i$, the flag cell C_{flag} of the sub-area, and all parent cells R_2 cell pairs with the same flag cell C_{flag}.

Algorithm 4. Divide the original attribute area.

```
1. Input: A_set
2. Output: A_sub_set, C_flag_set
3. Attr_sub ← {};
4. for a_i in A_set do
5.     if a_i satisfy Rule IV then
6.         C_flag_set.add(a_i);
7.         A_sub_set.add(Attr_sub);
8.         Attr_sub ← {};
9.     else
10.        Attr_sub.add(a_i);
11.    end if
12. end for
13. return A_sub_set, C_flag_set;
```

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

Algorithm 5. Match attribute area.

1. **Input**: A_{sub_set}, C_{flag_set}
2. **Output**: $R1_{set}$, $R2_{set}$
3. **for** $Attr_i$ in A_{sub_set} **do**
4. **for** a_i, a_j in $Attr_i$ **do**
5. **if** (a_i, a_j) satisfy Rule V **then**
6. $R1_{set}$.add(a_i, a_j);
7. **end if**
8. **end for**
9. **end for**
10. **return** $R1_{set}$, $R2_{set}$;

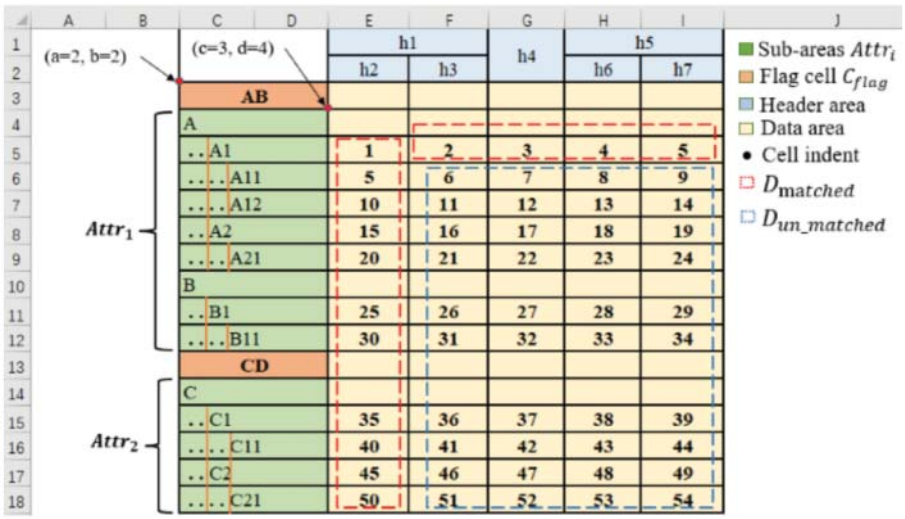

Figure 4. The original attribute area is divided into several sub-areas $Attr_i$, and the sub-areas are separated by the flag cell C_{flag}. In the Match D cell step, $D_{matched}$ and $D_{un\_matched}$ are the set of matched D cells and the set of unmatched D cells after the first round of matching process, respectively. In addition, a, b, and c, d are the row and column coordinates of the upper left corner and lower right corner of cell AB, respectively.

5. EXPERIMENTS

In this section, we will conduct experiments to evaluate the performance of HUSS, including the experimental setting, cell function analysis, and table structure analysis comparison, and the ablation studies.

5.1 Experimental Setup

5.1.1 Datasets

We evaluate the performance of HUSS on its cell function analysis and table structure analysis on five real-world spreadsheet datasets, containing cross-domain and different complex layout structures. Table 2

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

show the statistical information of spreadsheets in these datasets, including the number of tables, the proportion of hierarchical tables (i.e., the header area or the attribute area of the spreadsheet contains hierarchy), the average number of rows, and the average number of columns. The detail information of each dataset is as follows.

Table 2. Statistical information for five real-world spreadsheet datasets.

Datasets	Number of tables	The ratio of hierarchical tables	Average number of rows	Average number of columns
DeEx	457	82.1%	20.8	11.4
CIUS	268	67.8%	13.8	10.5
SAUS	210	79.1%	21.2	12.8
Troy200	200	93.7%	21.4	17.1
SAUS200	200	83.5%	19.9	12.1

- **DeEx** is collected from the DeExcelerator project and contains 457 annotated sheets.
- **CIUS** is primarily from the American Criminal Organizations (CIUS) database and contains 268 annotated sheets.
- **SAUS** is downloaded from the U.S. Census Bureau and contains 210 annotated sheets. Both CIUS and SAUS datasets are annotated in [2].
- **Troy200** contains 200 CSV files, containing 200 tables collected from 10 government statistics websites (most in English). We used an earlier version of it, which stores these tables with cell style characteristic information (bold, centered, indented) as spreadsheets.
- **SAUS200** is a random selection of 200 spreadsheets from the SAUS dataset, a 2010 statistical summary of the U.S. Bureau of National Statistics that includes 1369 Excel files downloaded from the U.S. Census Bureau.

5.1.2 Evaluation Metrics

Four popular metrics will be used to evaluate the performance of HUSS, namely Macro-F1, precision, recall, F1. Specifically, Macro-F1 will be used to evaluate the cell function analysis performance of HUSS. It is a common metric to evaluate the overall accuracy of multiple types. Precision, recall, and F1 will be used to evaluate the table structure parsing performance of HUSS.

5.1.3 Baselines

To demonstrate the effectiveness of HUSS on its cell function classification and its table structure analysis, the following baseline algorithms are chosen for comparisons, including RNN^{C+S} [11], RF [13], CRF [16], T_{ABBYXL} [18], $S_{ENBAZURU}$ [15].

- RNN^{C+S} is a method to embed the semantic and contextual information of a cell into a low-dimensional vector space. Then, a RNN model is trained from the vector space with the style characteristics of cells. Note that this method does not use the layout composition features of spreadsheets.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

- *RandomForest*(*RF*) uses a set of hand-crafted cell styles, including format, style, and layout characteristics of table cells, and trains a Random Forest (RF) classifier to classify cells. Note that it ignores cell semantics and context information.
- *ConditionalRandomField*(*CRF*) also uses the style, formatting, and layout characteristics of cells to train a classifier to classify cells. It also ignores cell semantics and context information.
- $T_{ABBY}XL$ is a rule-based method. It proposes a Cell Rule Language (CRL) to enforce defined rules, and then focuses on extracting the types of cell pairs, i.e., R_1, R_3, and R_4 contained in spreadsheets. However, it can only handle domain-specific spreadsheets.
- $S_{ENBAZURU}$ is based on an undirected graph model to only extract the types of cell pairs, e.g., R_1 in the top (header) and left (attribute) areas of a spreadsheet.

5.1.4 Experimental Environment

The proposed algorithm HUSS and all baselines in this paper are implemented in Python and run on a machine with Inter(R) Core(TM) i9-10900F CPU @2.80GHz.

5.2 Comparisons of Cell Function Classification Performance

In order to demonstrate the cell function classification performance of HUSS, we compare its performance with three baselines (i.e., RNN^{C+S} [11], RF [13], CRF [16]) described before, and our experimental results are shown in Table 3. Besides, since ECM is a general approach, we also apply it to further improve the performance of all three baselines. From Table 3, we can make following conclusions.

Table 3. Experimental results of cell function classification.

Datasets	Methods	Per-class F1			Macro-F1
		H	A	D	
DeEx	RF	73.1	44.3	97.9	71.8
	CRF	24.4	10.4	49.4	28.07
	RF+ECM	74.5	45.2	98.3	72.67
	CRF+ECM	25	11	52.1	29.37
	RNN^{C+S}	83.2	65.3	99.1	82.53
	HUSS	**84.1**	**66.2**	**99.2**	**83.17**
SAUS	RF	93.4	93.1	97.5	94.67
	CRF	89.3	86.6	97.6	91.17
	RF+ECM	94.0	93.7	98.1	95.27
	CRF+ECM	91.1	87.5	98.2	92.27
	RNN^{C+S}	95.1	95.0	98.0	96.03
	HUSS	**96.0**	**95.6**	**98.7**	**97.77**
CIUS	RF	98.2	94.2	99.0	97.13
	CRF	81.8	85.7	97.9	88.47
	RF+ECM	98.7	95.1	99.3	97.7
	CRF+ECM	85.2	89.1	98.2	90.83
	RNN^{C+S}	99.8	97.5	99.3	98.87
	HUSS	**99.9**	**97.8**	**99.5**	**99.07**

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

- Our HUSS performs the best, followed by RNN^{C+S}. CRF performs the worst, and RF performs much better than CRF.
- ECM does improve the performance of all three baselines. Specifically, HUSS performs better than RNN^{C+S}, CRF + ECM performs better than CRF, and RF + ECM performs better than RF. This demonstrates that ECM can effectively improve the performance of the three baselines on cell function classification.
- For a baseline algorithm with a poor cell function classification accuracy (e.g., CRF), the improvement effect of ECM is not obvious, while for a baseline algorithm with a high cell function classification accuracy (e.g., RNN^{C+S}), the improvement effect of ECM is obvious. This shows that ECM depends on the chosen baselines.

The Nemenyi [28] test is conducted to present the cell function classification performance comparisons between HUSS and all other algorithms in terms of macro-F1. In the Nemenyi tests, it is considered that a significant difference exists if the average ranks of two models differ by at least one critical difference (CD), which is calculated using a 5% significance level. For each model, its 30 experimental results (each algorithm is run 10 times on each of the three datasets) are statistically compared, as shown in Figure 5. The Nemenyi test results show that HUSS has better cell function classification performance.

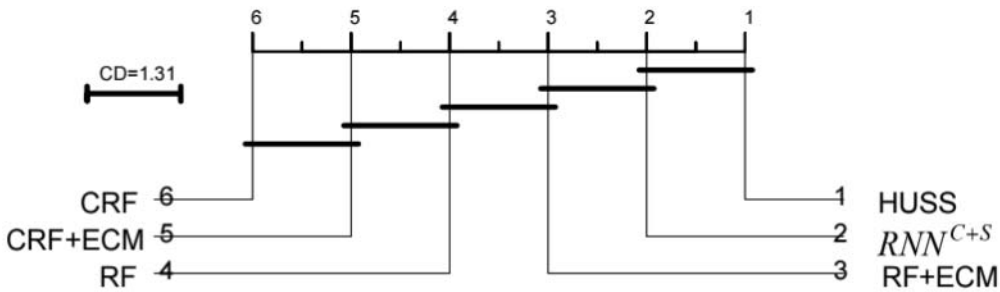

Figure 5. Nemenyi test results for the cell function classification performance in terms of macro-F1. The average rank of each algorithm is marked along the axis (the highest ranks to the most right). Models on the same level have similar cell function classification performance. The experimental results show that the cell function classification performance of HUSS outperforms all other algorithms.

5.3 Comparisons of the Table Structure Analysis Performance

In order to demonstrate the performance of HUSS on the table structure analysis, we compare HUSS with two baseline algorithms (i.e., $T_{ABBY}XL$ [18] and $S_{ENBAZURU}$ [15]) on two real-world spreadsheet datasets (i.e., Troy200 and SAUS200). Our experimental results are shown in Table 4. From Table 4, we can make following conclusions.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

Table 4. Experimental results on table structure analysis.

Datasets	Methods	SAUS200			Troy200		
		TabbyXL	Senbaruzu	HUSS	TabbyXL	Senbaruzu	HUSS
Precision	R_1	0.96	0.88	**0.98**	0.97	0.90	**0.99**
	R_2	0	0	**0.96**	0	0	**0.98**
	$R_3 + R_4$	0.96	0	**0.99**	0.98	0	**0.99**
Recall	R_1	0.78	0.88	**0.92**	0.93	0.89	**0.99**
	R_2	0	0	**0.92**	0	0	**0.97**
	$R_3 + R_4$	0.95	0	**0.98**	0.97	0	**0.98**
F1	R_1	0.86	0.88	**0.95**	0.95	0.89	**0.99**
	R_2	0	0	**0.94**	0	0	**0.97**
	$R_3 + R_4$	0.95	0	**0.99**	0.97	0	**0.98**

- HUSS outperforms both baseline algorithms (i.e., $T_{ABBY}XL$ and $S_{ENBAZURU}$) for extracting R_1 and R_2 relationship types of cell pairs. This demonstrates the effectiveness of the layout characteristics within the attribute area of spreadsheets. Both baselines ignore the layout characteristics within the attribute area of spreadsheets.
- It is obvious that $S_{ENBAZURU}$ can only extract one relationship type of cell pairs, and $T_{ABBY}XL$ can only extract three relationship types of cell pairs. However, our HUSS can extract four relationship types of cell pairs. This shows that HUSS can analyze the spreadsheet structure more accurately and broadly.

The Nemenyi [28] test of HUSS and all other algorithms in table structure analysis performance in terms of precision, recall, and F1 was carried out. For each model, its 30 experimental results (each algorithm was run 10 times on each of the three metrics for each dataset) are statistically compared, as shown in Figure 6. The experimental results demonstrate that the table structure analysis performance of HUSS is significantly better than all other comparison algorithms.

5.4 Ablation Studies

In this section, we conduct ablation studies to investigate the effectiveness of our proposed error correction mechanism (ECM) and the five types of heuristic rules for HUSS for understanding the semantic structure of spreadsheets. Specifically, we construct six variants of HUSS, i.e., HUSS-w/o ECM (removing ECM), HUSS-w/o Rule I (removing Rule I), HUSS-w/o Rule II (removing Rule II), HUSS-w/o Rule III (removing Rule III), HUSS-w/o Rule IV (removing Rule IV), and HUSS-w/o Rule V (removing Rule V). Our experimental results on two real-world spreadsheet datasets (i.e., Troy200 and SAUS200) are shown in Table 5. From Table 5, we can make following conclusions.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

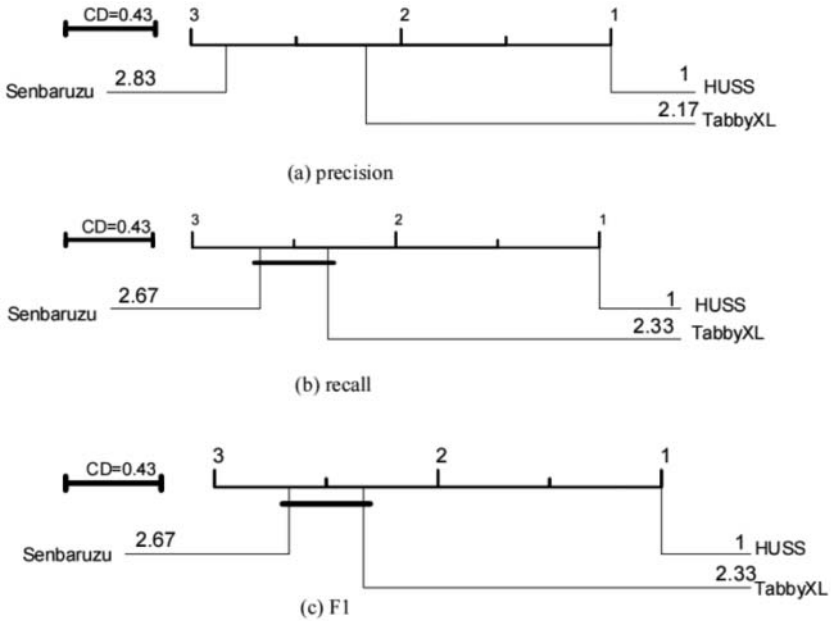

Figure 6. (a), (b), (c) are the Nemenyi test results of table structure analysis. performance in terms of precision, recall, and F1, respectively.

Table 5. Ablation experimental results of six variants of HUSS.

Datasets	Methods	Per-class F1			Macro-F1
		R_1	R_2	$R_3 + R_4$	
SAUS200	HUSS-w/o ECM	0.8141	0.8292	0.9133	0.8522
	HUSS-w/o Rule I	**0.9524**	**0.9416**	0	0.6313
	HUSS-w/o Rule II	**0.9524**	**0.9416**	0.2804	0.7248
	HUSS-w/o Rule III	0.7424	0.7182	**0.9926**	0.8177
	HUSS-w/o Rule IV	0.7940	0.8890	**0.9926**	0.8919
	HUSS-w/o Rule V	0.3240	0.2796	**0.9926**	0.5312
	HUSS	**0.9524**	**0.9416**	**0.9926**	**0.9622**
Troy200	HUSS-w/o ECM	0.8598	0.8409	0.9062	0.8690
	HUSS-w/o Rule I	**0.9962**	**0.9767**	0	0.6576
	HUSS-w/o Rule II	**0.9962**	**0.9767**	0.3714	0.7814
	HUSS-w/o Rule III	0.7376	0.7465	**0.9896**	0.8246
	HUSS-w/o Rule IV	0.8446	0.9112	**0.9896**	0.9149
	HUSS-w/o Rule V	0.2917	0.3085	**0.9896**	0.5299
	HUSS	**0.9962**	**0.9767**	**0.9896**	**0.9875**

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

- When the error correction mechanism ECM is removed, the F1 value of HUSS on extracting all four relationship types of cell pairs drops. This again proves the effectiveness of the ECM mechanism.
- When Rules I and II are removed, the F1 value of R_1 and R_2 of cell pairs remains unchanged, but the F1 value of R_3 and R_4 of cell pairs is drastically reduced. This is because Rule I and II are mainly the rule sets for the data cells of spreadsheets, which do not affect the discovery of parent-child and sibling relationships of cell pairs in the header and attribute areas. However, Rule I and II help discover R_3 and R_4 of cell pairs in spreadsheets.
- When Rules III, IV, and V are removed individually, the F1 value of R_3 and R_4 of cell pairs remains unchanged, but the F1 value of R_1 and R_2 of cell pairs is reduced. This is because Rules III, IV, V mainly set the rules for the header and attribute areas of spreadsheets. They do not affect the cells in the data area. This proves the effectiveness of Rules III, IV, V for HUSS to extract R_1 and R_2 of cell pairs.

6. CONCLUSION

Spreadsheets are usually organized in a semi-structured manner and contain complex hierarchies, which make machines difficult to understand their semantic structures. In this paper, we proposed a heuristic algorithm for understanding the semantic structure of spreadsheets (HUSS). HUSS contains two main components for cell function analysis and table structure analysis respectively. For the cell function analysis, we proposed an error correction mechanism (ECM) based on the cell function classification model [11] and the layout composition characteristics of spreadsheets. Our experimental results showed that it can effectively detect and relabel misclassified cells and further improve the accuracy of cell function classification. For the table structure analysis, we proposed five types of heuristic rules to extract four different relationship types of cell pairs in spreadsheets by using the functional types, styles, spatial positions of cells, and the layout characteristics of the attribute areas of spreadsheets. Our experimental results showed that HUSS outperforms the baselines and can extract all four relationship types of cell pairs.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China under Grants (Nos. 62120106008, 61806065, 61906059, 62076085, 91746209 and 62076087), and the Fundamental Research Funds for the Central Universities (No. JZ2020HGQA0186).

REFERENCES

[1] Wang, Z., Dong, H.Y., Jia, R., et al.: TUTA: Tree-based transformers for generally structured table pretraining. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1780–1790 (2021)

[2] Lehmberg, O., Ritze, D., Meusel, R., et al.: A large public corpus of web tables containing time and context metadata. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 75–76 (2016)

Downloaded from http://direct.mit.edu/dint/article-pdf/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

[3] Kappelman, L.A., Thompson, J.P., Mclean, E.R.: Converging enduser and corporate computing. Communications of the ACM, pp. 79–92 (1993)
[4] Lehmberg, O., Ritze, D., Ristoski, P., et al.: The mannheim search join engine. Journal of Web Semantics, pp. 159–166 (2015)
[5] Zhang, L., Zhang, S., Balog, K.: Table2vec: Neural word and entity embeddings for table population and retrieval. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1029–1032 (2019)
[6] Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1422–1432 (2015)
[7] Du, L., Gao, F., Chen, X., et al.: TabularNet: A neural network architecture for understanding semantic structures of tabular data. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 322–331 (2021)
[8] Sun, K.X., Rayudu, H., Pujara, J.: A hybrid probabilistic approach for table understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4366–4374 (2021)
[9] Zhang, Y.K., Xiao, L., Dong, H.Y., et al.: Semantic table structure identification in spreadsheets. In: Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 283–295 (2021)
[10] Hurst, M.: The interpretation of tables in texts PhD thesis. University of Edinburgh. School of Cognitive Science, Informatics (2000)
[11] Gol, M.G., Pujara, J., Szekely, P.: Tabular cell classification using pre-trained cell embeddings. In: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, pp. 230–239 (2019)
[12] Chen, Z., Cafarella, M.: Automatic web spreadsheet data extraction. In: Proceedings of the 3rd International Workshop on Semantic Search over the Web, pp. 1–8 (2013)
[13] Koci, E., Thiele, M., Romero Moral, O., et al.: 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, vol. 1, pp. 77–88 (2016)
[14] Adelfio, M.D., Samet, H.: Schema extraction for tabular data on the web. In: Proceedings of the VLDB Endowment, pp. 421–432 (2013)
[15] Chen, Z., Cafarella, M., Chen, J., et al.: Senbazuru: A prototype spreadsheet database management system. In: Proceedings of the VLDB Endowment, vol. 6, pp. 1202–1205 (2013)
[16] Chen, Z., Cafarella, M.: Integrating spreadsheet data via accurate and low-effort extraction. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1126–1135 (2014)
[17] Chen, Z., Dadiomov, S., Wesley, R., et al.: Spreadsheet property detection with rule-assisted active learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 999–1008 (2017)
[18] Shigarov, A.O., Mikhailov, A.A.: Rule-based spreadsheet data transformation from arbitrary to relational tables. Information Systems, vol. 71, pp. 123–136 (2017)
[19] Shigarov, A.O., Paramonov, V.V., Belykh, P.V., et al.: Rule-based canonicalization of arbitrary tables in spreadsheets. In: International Conference on Information and Software Technologies. Springer, pp. 78–91 (2016)
[20] Paramonov, V., Shigarov, A., Vetrova, V.: Rule driven spreadsheet data extraction from statistical tables: case study. In: International Conference on Information and Software Technologies. Springer, pp. 84–95 (2021)
[21] Pujara, J., Rajendran, A., Ghasemi-gol, M., et al.: A common framework for developing table understanding models. In ISWC Satellites, pp. 133–136 (2019)

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

[22] Bonfitto, S., Casiraghi, E., Mesiti, M.: Table understanding approaches for extracting knowledge from heterogeneous tables. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 11(4), e1407 (2021)
[23] Koci, E., Kuban, D., Luettig, N., et al.: Xlindy: Interactive recognition and information extraction in spreadsheets. In Proceedings of the ACM Symposium on Document Engineering, pp. 1–4 (2019)
[24] Koci, E., Thiele, M., Romero, O.: A genetic-based search for adaptive table recognition in spreadsheets. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE, pp. 1274–1279 (2019)
[25] Dong, H., Liu, S., Han, S., et al.: Tablesense: Spreadsheet table detection with convolutional neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 69–76 (2019)
[26] Dou, W., Han, S., Xu, L., et al.: Expandable group identification in spreadsheets. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 498–508 (2018)
[27] Shigarov, A., Khristyuk, V., Mikhailov, A., et al.: Tabbyxl: Rule-based spreadsheet data extraction and transformation. In: International Conference on Information and Software Technologies, Springer, pp. 59–75 (2019)
[28] Demar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, pp. 1–30 (2016)
[29] Wu, X.D., Chen, H., Bu, C.Y., et al.: HUSS: A heuristic method for understanding the semantic structure of spreadsheets. In: 2022 International Conference on Knowledge Graph (ICKG), IEEE, (2022)

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

AUTHOR BIOGRAPHY

Xindong Wu is Director and Professor of the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, China, and Senior Research Scientist in the Research Institute of Artificial Intelligence at Zhejiang Lab, Hangzhou, China. His research interests include data mining, knowledge engineering, and web information exploration. He received his Ph.D. in artificial intelligence from the University of Edinburgh. He is the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), and the Editor-in-Chief of Knowledge and Information Systems. He is Fellow of IEEE and the AAAS, and a foreign member of the Russian Academy of Engineering.

Hao Chen is a graduate student of the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China) and the School of Computer Science and Information Engineering at the Hefei University of Technology, Hefei, China. He received the BS degree from Anhui University of Finance and Economics. His research interests include semantic understanding of spreadsheets.

Chenyang Bu received his Ph.D. degree from the University of Science and Technology of China (USTC). He is currently an associate professor at the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei 230601, China. He serves as a reviewer for several international journals including 10+ IEEE/ACM Transactions. His main research interests include knowledge graph construction and applications, as well as automated graph learning with evolutionary algorithms.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024

Shengwei Ji received his Ph.D. degree from the Hefei University of Technology in 2022, and B.S. degree from Changan University in 2015. He is currently an assistant professor with Hefei University. He has published several peer-reviewed papers in prestigious journals and top international conferences including IEEE TETC, ACM TIST, and IEEE ICDCS. His research fields lie in local graph learning, distributed computing, and knowledge graph.

Zan Zhang is an associate professor of the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China) and the School of Computer Science and Information Engineering at the Hefei University of Technology, Hefei, China. He received his Ph.D. degree in Computer Science from the Hefei University of Technology, China (2018). He was a visiting PhD student at the University of South Australia from 2015 to 2017. His research interests include causal inference and weakly supervised learning.

Victor S. Sheng is an Associate Professor of Computer Science and the founding Director of Data Analytics Lab (DAL) at Texas Tech University. He received his Master's degree in Computer Science from the University of New Brunswick, Canada, in 2003, and his Ph.D. degree in Computer Science from Western University, Ontario, Canada, in 2007. His research interests include data science, natural language processing, machine learning and data mining, crowdsourcing, and related applications in business, industry, medical informatics, and software engineering. He was an Associate Professor at the University of Central Arkansas, and an Associate Research Scientist and NSERC Postdoctoral Follow in Information Systems at Stern Business School, New York University. Dr. Sheng is a senior member of IEEE and a lifetime member of ACM.

Downloaded from http://direct.mit.edu/dint/article-pdf/5/3/537/2158253/dint_a_00201.pdf by guest on 23 May 2024