DECO: A Dataset of Annotated Spreadsheets for Layout and Table Recognition

Elvis Koci*†, Maik Thiele*, Josephine Rehak*, Oscar Romero†, Wolfgang Lehner*

*Fakultät Informatik
Technische Universität Dresden
Dresden, Germany
name.surname@tu-dresden.de

†Departament d'Enginyeria de Serveis i Sistemes d'Informació
Universitat Politècnica de Catalunya-BarcelonaTech
Barcelona, Spain
{ekoci, oromero}@essi.upc.edu

***Abstract*—This paper presents DECO (Dresden Enron COrpus), a dataset of spreadsheet files, annotated on the basis of layout and contents. It comprises of 1,165 files, extracted from the Enron corpus [1]. Three different annotators (judges) assigned layout roles (e.g., Header, Data, and Notes) to non-empty cells and marked the borders of tables. Files that do not contain tables were flagged using categories such as Template, Form, and Report. Subsequently, a thorough analysis is performed to uncover the characteristics of the overall dataset and specific annotations. The results are discussed in this paper, providing several takeaways for future works. Furthermore, this work describes in detail the annotation methodology, going through the individual steps. The dataset, methodology, and tools are made publicly available, so that they can be adopted for further studies. DECO is available at: https://wwwdb.inf.tu-dresden.de/research-projects/deexcelarator/**

***Index Terms*—Spreadsheet, Dataset, Enron, Corpus, Annotation, Recognition, Layout, Table, Templates**

I. Introduction

Spreadsheets are the tool-of-choice for many different settings, such as business, open data, and academia. They are intuitive to use, with a low entrance barrier. Nonetheless, they provide a broad range of advanced functionalities, enabling data collection, transformation, analysis, and reporting. For these reasons, among other, spreadsheets have become very popular with novices and professionals alike.

As a consequence, a large volume of data can be found in spreadsheet documents. Of particular interest are data coming in tabular form, since they provide concise and to large extend structured information. There are clear benefits from automatically recognizing and processing such data. A typical example is that of integrating with other sources and systems. This can boost reusability and provide valuable data for business intelligence tasks. Other benefits are better governance for spreadsheets and improved user (employee) experience.

However, tables in spreadsheets are often intermingled with formatting artifacts, textual metadata, ad-hoc calculations, and floating objects (e.g., shapes, charts, and pictures). Moreover, contents can be arranged in arbitrary ways, depending on the user preferences. Thus, automatic table recognition and analysis is rather challenging in spreadsheets.

In literature, there is a considerable number of proposed approaches, mentioned in Section II. However, we notice that these works use different datasets for their evaluation. These datasets vary significantly in composition, size, and annotation methodology. On top of that, the majority of works do not make their annotations publicly available. Thus, it is difficult to confirm and compare their findings. Lastly, these works typically study spreadsheets from the Web, which are easily accessible (e.g., open data platforms). Enterprise spreadsheets are much less studied, since companies are reluctant to share their internal documents. A higher degree of complexity is expected from such spreadsheets, but this is yet be confirmed.

To address these problems, we propose DECO, a large-scale and ready-to-use dataset of real-word spreadsheets, annotated on the basis of layout and contents. DECO is able to confirm or invalidate previous assumptions, and can be used as a benchmark within the research community. Specifically, we have annotated a sample of $1,165$ documents, extracted from the Enron corpus [1]. We perform a thorough analysis of the annotated documents by investigating various aspects, such as the density and arrangement of contents, and the usage of specific formatting artifacts. In particular, this study provides valuable insights on the characteristics of tables in spreadsheets. For example, we check the presence of hierarchies in tables, such as nested Headers. Moreover, we study how often tables contain structural "anomalies", such as empty cells/rows/columns. Lastly, besides the dataset, we provide a comprehensive description of our annotation methodology, and tools that can be adapted for further studies.

The subsequent parts of this paper are organized as follows: We discuss the related work in Section II. We outline the methodology used for the creation of the dataset, in Section III, Furthermore, in Section IV, we describe the DECO dataset with statistics. We conclude this paper with Section V.

II. Related Work

We find multiple spreadsheet corpora, in the literature. These have almost entirely focused on Microsoft Excel files, as it is the most popular spreadsheet application. Furthermore,

This work is supported by the German Federal Ministry of Education and Research (BMBF, 01IS14014A-D), by funding the competence center for Big Data "ScaDS Dresden/Leipzig"

these files are typically crawled from the Web, where a considerable amount of spreadsheets is publicly available.

Euses [2] has served the spreadsheet community for a while. It was created with the help of search engines, issuing queries containing keywords such as “financial” and “inventory”, and file type “.xls”. Overall, it comprises of $4,498$ unique spreadsheets, organized into categories (folders) based on the used keywords. The more recent Enron corpus [1] contains $15,770$ spreadsheets, extracted from the Enron email archive[1]. This corpus is unique, for its exclusive view on the use of spreadsheets in enterprise settings. All the files were used internally by Enron company, from August 2000 to December 2001. Overall, these files relate to one or more of the 130 distinct employees, from the email records. Another recent corpus is Fuse [3], which comprises of $249,376$ unique spreadsheets, extracted from Common Crawl[2]. Each spreadsheet is accompanied by a JSON file, which includes NLP token extraction and metrics related to the use of formulas.

So far, these three corpora have been used by researchers viewing spreadsheets from a software engineering perspective. Formula error detection and debugging [4], [5], but also usage, life-cycle, modeling, and governance of spreadsheets [6]–[8] are important research subjects within this community.

There are works that report on annotated spreadsheet files. Here, we mention those focusing on table recognition and layout inference. Even though these works have made part of the original files available, the related annotations are not public. Furthermore, these works lack a proper discussion of their annotation methodology and tools. Thus a direct comparison is currently not possible.

Chen and Cafarella, at [9], crawled $410,554$ Microsoft Excel files using the ClueWeb09[3] dataset. Out of these files, 100 were annotated at the row level, using one of the following layout roles: *Title, Header, Data*, and *Footnotes*. In a subsequent work [10], Chen and Cafarella present SAUS R200, a sample of 200 spreadsheets from the 2010 Statistical Abstract of the United States. Moreover, the same paper includes WEB R200, an extension of the dataset from [9]. Both WEB R200 and SAUS R200 were annotated to capture header and data hierarchies, found in spreadsheet tables. In their latest work [11], they discuss WEB400. This dataset comprises of 400 spreadsheets, additionally annotated with table properties, such as aggregation rows/columns and nested tables.

Adelfio and Samet [12] simultaneously deal with tables in spreadsheets and HTML documents. The authors annotated 1117 Microsoft Excel files and 1204 HTML pages, crawled from the Web. Similar to [9], they annotate at the row level. However, they instead use 7 layout labels, which are more specialized than [9].

Other works, such as [13], [14], do not rely on annotations, but rather work with small datasets (< 50 files). The performance is manually assessed per file.

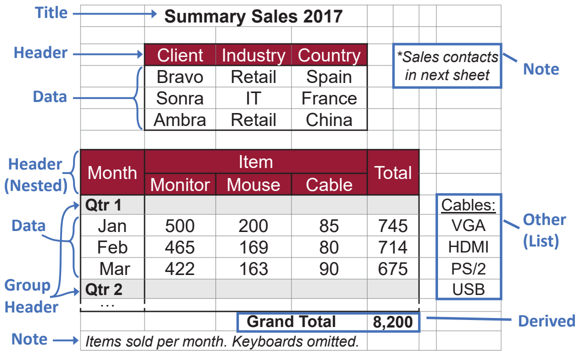
Koci et al. [15] use a dataset of 216 annotated spreadsheets. Unlike aforementioned works, the annotations and tools are made public. However, this dataset is very diverse, mixing files from three corpora: Enron [1], Euses [2], and Fuse [3]. Instead, in this work, we focus on business spreadsheets.

III. Methodology

In this section we present the tools and methods that were used to create the DECO dataset.

A. Annotation Labels

We define two sets of annotation labels, at the cell and sheet level. The former provides layout roles, such as *Title*, *Header*, and *Data*, which are attached to non-empty cells of the sheet. Additionally, this set includes the label *Table*, which describes a region (group) of annotated cells. The second set is used for sheets that do not contain table structures. Therefore, we use labels such as *Template* and *Report*, to describe these cases.



Client	Industry	Country
Bravo	Retail	Spain
Sonra	IT	France
Ambra	Retail	China

Month	Item			Total
	Monitor	Mouse	Cable	
Qtr 1				
Jan	500	200	85	745
Feb	465	169	80	714
Mar	422	163	90	675
Qtr 2				
...				
		Grand Total		8,200

Cables:
VGA
HDMI
PS/2
USB

Fig. 1. Cell Annotation Labels

1) Cell Labels: As shown in Figure 1, we define seven roles for non-empty cells: *Data*, *Header*, *Derived*, *GroupHeader*, *Title*, *Note*, and *Other*. We follow closely the Wang model [16], and the labels proposed by previous work [9], [12], [15].

The basic ingredients for tables are Headers and Data. The former give names to columns, describing the values below them. Headers can be nested occupying several consecutive rows, as shown in Figure 1. Data cells are the main payload of values in a table. They follow the structure defined by Headers.

Derived cells are aggregations of Data, such as sum, product, and average. Here, we specifically focus on aggregations “interrupting” the Data rows. In other terms, Derive act as subtotals or grand totals. It is important to distinguish such aggregations from the rest, since they clearly affect the structure, coherency, and shape of a table. On the contrary, aggregations in columns tend to have a lower impact. Therefore, we annotate them simply as Data (see Figure 1).

GroupHeaders (also referred to as *GHead*) are reserved for hierarchical structures on the left of a table. In such structures, values in column/s are nested, implying parent-child relationships. When spotted, we annotate parents as GroupHeaders, while children as Data.

Titles and *Notes* provide additional information, effectively assisting at the understanding of sheet contents. Titles give a

[1] http://info.nuix.com/Enron.html
[2] http://commoncrawl.org/
[3] http://lemurproject.org/clueweb09.php/

name to specific sections (such as a table), or to the sheet as a whole. Notes provide comments and clarifications, which again can apply globally or locally. Typically, Notes take the form of a complete or almost complete sentences. On the other hand, Titles can consist of just a single word.

The label *Other* is placeholder for everything else, not fitting to the aforementioned cell labels. Additionally we annotate as Other regions that do not comply with our definition of a table (see Section III-A2). For instance, these can be "Data" values that are not preceded by a Header row/column. Moreover, Other is used for regions containing key-value pairs. A typical example are parameters used for calculations in the sheet.

Finally, a *Table* is annotated with the minimum bounding rectangle (MBR) enclosing all non-empty cells that compose it. Intuitively, some cell labels (i.e., Header, Data, and Group-Header) are only found inside table annotations. With regards to Derived, they are primarily found in tables. However, when Derived are used to aggregate Data from multiple tables, we leave them outside. Titles and Notes can relate to multiple sections of the sheet. Thus, they are not integral part of table annotations. Lastly, being a versatile label, Other can be found both in and outside of table borders.

2) Sheet Labels: Besides cells, we annotate non-empty sheets of a spreadsheet file. Here, we focus on those that do not contain tables, which we refer to as *Not-Applicable (N/A)*. These kind of sheets are flagged with one of the following labels: *Form/Template*, *Report/Balance*, *Chart*, *List*, *NoHeader*, and *Other*. We define Form/Templates as sheets intended to be re-used again for similar tasks (e.g., collecting data, performing specialized calculations). Therefore, they are usually accompanied by instructions on how to use them. They might be filled with artificial (example) values or not. Balance/Report are typically used to summarize financial performance. They might report on company's assets, liabilities, and shareholders' equity. Often, content in these sheets is not organized in a table-like fashion. Next, the label List is used for sheets that have values organized in a single column. The label NoHeader applies when we do not find Header cells in the sheet, even though there are values organized in multiple rows and columns. Charts are sheets that contain plots/diagrams and the source values (not described by Headers). Finally, similarly to cell annotations, we introduce the label Other, for sheets that do not match the aforementioned labels.

B. Annotation Tool

For this work, we extended the annotation tool introduced at [15]. The updated version[4] supports the proposed annotation labels (see Section III-A). Moreover, it runs background checks, enforcing our annotation logic (as outlined in Section III-C), and warning the users when a potential false step is about to happen.

The tool prevents any alteration of the original formatting and contents of the loaded file. Cells are annotated by first selecting a (rectangular) region in a sheet, and subsequently a label from the designated sub-menu. Likewise, the active sheet itself can be annotated using options from the menu.

User annotations are saved inside the loaded file, in designated hidden sheets, created by the tool. The current status of the file and its sheets are recorded, as well. The status remains *In Progress*, unless the user indicates (from the menu) that the file/sheet is either *Completed* or *Not-Applicable*. Both annotations and statuses can be exported as illustrated here[4].

All non-pending files (i.e., Completed and Not-Applicable) are organized in two folders, by the annotation tool. Those that have a sheet with table annotation/s end up in the *completed* folder, while the rest go to the *not-applicable* folder. The latter is divided further into sub-folders, which correspond to the N/A labels. This with exception to *multi-na*, which holds files with multiple N/A sheets, but flagged with a different label.

C. Annotation Task

The annotators task is to inspect each file for table/s. When a file contains no table, the sheets must be flagged with the appropriate N/A label (see Section III-A2). Subsequently, the file status is changed to Not-Applicable, before saving it. If there are tables in the file, judges determine the first sheet (FS) among those having one, following the tabs from left to right. Judges annotate all non-empty cells and tables in FS, prior to changing its status to Completed. The sheets that come before FS are flagged with the appropriate N/A label, while those that follow FS are ignored (i.e., maximum one Completed sheet per file). To conclude the task, also the status of the file is changed to Completed, and subsequently saved.

D. Pre-selection of Files

1) Original Dataset: The Enron corpus [1] consist of 15,770 spreadsheet files, extracted from Enron email archive. They were created during a period spanning from August 2000 to December 2001, recording various activities of the Enron corporation. The original email records were organized into 130 folders, one per employee. Enron corpus has followed the same logic, grouping the extracted spreadsheets by employee. It is not made clear if these employees are the original authors of the files, or just the authors of the emails. Regardless, in this work we follow the same pattern, keeping a close link between employees and files.

2) Initial Filtering: Files of the original dataset underwent an initial filtering, after which a considerable number of them were omitted. The maximum size of the file was limited to 5MB. Additionally, files with macros were filtered out. Moreover, we omitted those having broken external links to other files. Furthermore, we inspected the encoding, keeping only those having character set ANSI (Windows-1252)[5]. This makes it more probable that the selected files have English string values. In addition, we filtered out files that have similar name (Levenshtein distance ≤ 4) to one of those already selected. This step eliminated the biggest chunk of files, but also decreased the chance of having duplicates or

[4] https://wwwdb.inf.tu-dresden.de/research-projects/deexcelarator/

[5] The default encoding, for US based systems

near-duplicates in the reduced dataset. Lastly, some files were eliminated due to exception occurring while processing them with Apache POI[6] v3.17.

After filtering, the reduced dataset consists of 5,483 files, and 128 distinct employees. While there is a minimum of one file per employee, we find 15 employees with more 100 files.

E. Annotation Phases

The judges were three students in STEM fields. They had various degree of familiarity with Excel, prior to this study. To avoid any influence whatsoever, briefing and communication with the judges was handled individually.

The annotation process was organized into three phases: training, agreement assessment, and independent annotation.

1) Training Phase: The aim of training phase was to familiarize the judges with the tool, annotation task, and labels. They were given a written description of the task, annotated examples, and a small sample of files to practice with.

2) Agreement Phase: In the second phase, we performed an assessment of agreement between the judges that participated in the creation of DECO dataset. The aim was to ensure high degree of common understanding, prior to the third phase. Note, this is crucial for a dataset created by multiple independent judges, with different initial knowledge.

Initially, judges received a dataset of 128 files (one random file per employee). After the initial annotation, the agreement between judges was assessed, for the first time. Subsequently, we instructed them to review files in which there were substantial disagreements. The disagreements were described at the cell, sheet, and file level. Note, that some disagreements were due to negligence. Thus, another purpose of this phase was to fix trivial mistakes. Regardless, it was up to the judges to decide if to change their initial annotations or not.

Following the revisions, the agreement was assessed again. The results of this second assessment are presented in Table I. We used two metrics: Fleiss' Kappa [17] and Agreement Ratio. The former is a statistical measurement for the reliability of agreement between multiple judges. The latter captures the % of (annotated) items for each the judges agree. With regards to files and sheets, for this assessment we reduced the votes to either Not-Applicable or Completed. While for cells, judges vote with one of the seven available annotation labels.

As shown below, the agreement and its reliability are substantial, when studied at file, sheet, and cell level. Moreover, we measured the agreement[7] individually for each cell label. For Data and Header the agreement ratio is notably high. Additionally, for Derived judges have a significant agreement. For the rest, it is much lower. These results imply that labels closely associated with tables are more natural to the judges.

TABLE I
AGREEMENT ASSESSMENT

	Files	Sheets	Cells
Fleiss	0.77	0.72	0.86
Ratio	0.90	0.97	0.98

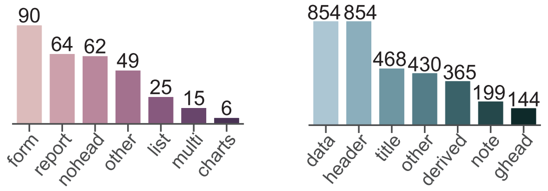
	Data	Header	Derived	Title	Other	GHead	Notes
Ratio	0.98	0.89	0.70	0.53	0.41	0.40	0.20

At the last step, we inspected the annotations manually, in order to better identify reasons behind the disagreements. For each judge, we determined cases where they had used labels incorrectly. Afterwards, we discussed these individually with the judges, clarifying any misunderstandings. Moreover, we asked them to correct their annotations accordingly. Again, these corrections were examined, confirming that understanding had indeed improved, among the judges.

3) Independent Annotation Phase: In the final phase, the judges were provided with an individual dataset and worked under minimum supervision. Files used previously in the agreement phase, were excluded from this phase. Each judge got a stratified sample of the remaining Enron dataset, covering files from 120 to 122 Enron employees.

IV. ANNOTATION STATISTICS

The dataset consists of $1,165$ annotated files. Out of these, 311 were marked as Not-Applicable (i.e., do not contain a table), while the rest, 854, were annotated at the cell level.



(a) Not-applicable Files (b) Sheet Count per Label

Fig. 2. Annotation in Numbers

A. Not Applicable Files

Figure 2.a shows the number of files per N/A sub-folder (see Section III-B). The dataset has a considerable amount of files with Form/Templates sheets. This subset can be of good use to the software engineering branch of spreadsheet research (see Section II). Additionally, we notice a high presence of NoHeader files. This suggest that occasionally users might omit headers. We choose to see these Header-less regions of "Data" values as non-valid tables. This complies with the current approaches to table recognition and analysis in spreadsheets, such as [9], [12], [18], which largely depend on the context provided by headers.

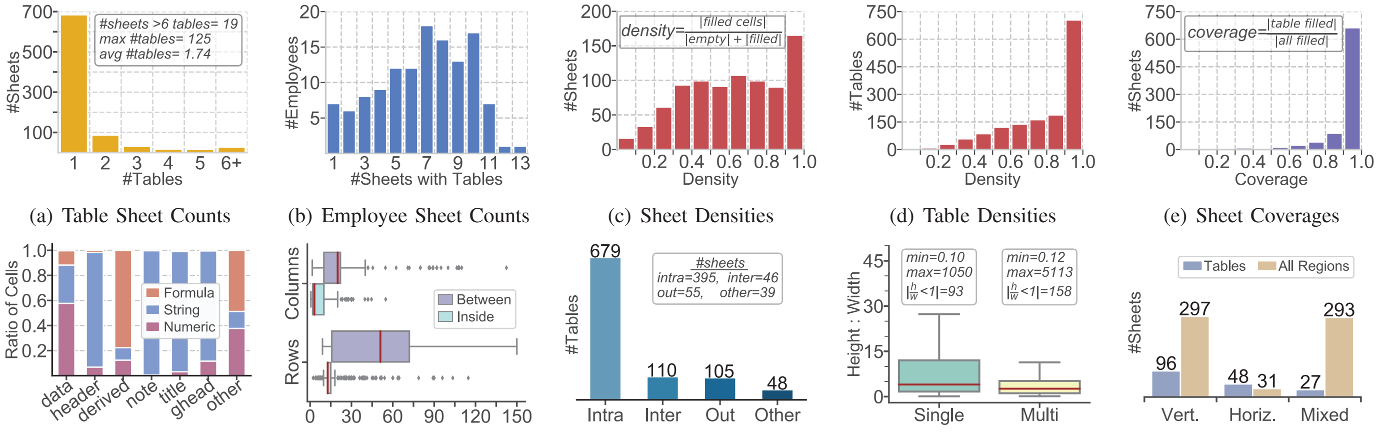
Takeaway 1: Spreadsheet users rely extensively on implicit information. At times, this might lead to omitted headers.

B. Sheets with Tables

Hereinafter, we discuss the sheets containing cell and table annotations. Besides other, we put into test claims and assumptions from related work. In Figure 2.b, we show for each cell label the number of sheets that have it. Data and Header are present in all the sheets, as expected. Also, we observe high

[6] https://poi.apache.org/

[7] Fleiss' Kappa omitted for cell labels, due to skewed vote distribution



(a) Table Sheet Counts

(b) Employee Sheet Counts

(c) Sheet Densities

(d) Table Densities

(e) Sheet Coverages

(f) Content Type Ratios

(g) Table Gaps

(h) Formula References

(i) Height to Width Ratio

(j) Arrangements

Fig. 3. Annotations Statistics for Sheets Containing Tables

occurrence of Titles, which seem to be preferred over Notes. We find Derived in ca. 43% of the sheets. This confirms our expectations, since 58% of the original Enron files contain a formula (any kind) [1]. GroupHeaders (ghead), i.e., left column hierarchies, are fairly common (17%). Moreover, 32% of the sheets contain nested headers (i.e., top hierarchies). These findings call for specialized approaches to handle hierarchical-style metadata in spreadsheets, such as [10].

Takeaway 2: Hierarchies on the top and left of annotated tables are common in business spreadsheets.

Furthermore, in Figure 2.b, we observe that more than half of these sheets contain Other. This implies that spreadsheet contents are highly diverse. Thus, even more labels than the ones considered in this work are needed to describe spreadsheets contents with high precision.

Takeaway 3: Contrary to previous assumptions, we notice substantial variety for layout and contents in spreadsheets.

In Figure 3.a, we summarize the number of table annotations per sheet. In total, there are 1,487 annotated tables. The vast majority, 683 sheets, contain only one table. Nevertheless, there are 171 sheets with two or more tables. Also, there are few extreme outliers with 34, 49, or even 125 tables.

Takeaway 4: The simplistic view of one table per sheet, often, does not hold for business spreadsheets.

Lastly, we examine the number of sheets (containing table annotations) per employee. Figure 3.b summarizes our analysis. We observe that the vast majority of employees contribute to 5 to 10 annotated sheets. These leaves adequate space for future user studies, which can reveal interesting pattern of spreadsheet usage in business settings.

C. Content Statistics

In this section, we study various aspects of spreadsheet contents. For our first analysis, we use the density and coverage metrics, proposed in [19]. *Density* captures the concentration of non-empty (*filled-in*) cells. We measure density for each individual table annotation, as well as for the complete used area of the sheet. The latter is the minimum bounding rectangle that encloses all *filled-in* cells of the sheet. Intuitively, the lower is the number of empty cells the higher is the density. *Coverage*, reports the ratio of filled-in cells in the sheet located inside the tables annotations.

Figures 3.c-e show the distributions of these measurements. The histograms consist of 10 bins (intervals), each having a width of 0.1. We observe that sheet densities vary extensively. Partially, this comes due to cells located outside tables, such as Titles and Notes. However, as shown in Figure 3.d, we notice a considerable number of sparse tables, as well. In Figure 3.e, we can observe that tables hold the largest portion of filled-in cells, for the majority of sheets.

Takeaway 5: The density of information in spreadsheets varies extensively. We often see sparse tables.

Furthermore, we discuss the distribution of content types per cell label. The results are shown in Figure 3.f. As anticipated, for Headers, Titles, Notes, and GroupHeaders we observe mostly string values. Additionally, we find a considerable amount of strings in Data cells (ca. 30%). For Derived we notice a large portion of numeric values, which suggests that occasionally users set the aggregation values manually (i.e, without using formulas). Finally, most of the cells annotated as Other are non-strings. This implies that Other might be closer to Data and Derived, rather than to the remaining labels.

Another analysis is that of empty rows/columns (referred to as gaps). Current approaches naïvely see such gaps as separators of tables. To test this assumption, we measured the height/width of adjacent empty rows/columns, inside and between tables of the sheet. Figure 3.g shows the distribution for these values (outliers >150 are omitted). Contrary to previous assumptions, row/column gaps are often found inside tables (respectively in 546 and 240 tables). Moreover, especially for column gaps, we notice a significant overlap for the distributions *inside* and *between*. Thus, the size of the gap is not always informative as to infer its purpose (i.e., being a table separator or just a formatting artifact).

Takeaway 6: We find empty row/column gaps inside tables. To distinguish them from those found between tables, an analysis that goes beyond their width/height is needed.

We conclude this section with a study of formulas found inside the annotated tables. Here, we focus on the references of these formulas. The intention is to capture the dependencies of table contents to the rest of the sheet or file. We observe that most of the table formulas refer to cells inside the same table (*Intra*). Additionally, we notice external references, in less that 8% of the tables. These can be from one table to another (*Inter*), referring to cells found outside the table (*Out*), or references to other sheets of the same file (*Other*).

Takeaway 7: Table content might depend on values found outside its borders; infrequently, this reside in other sheets.

D. Structural Statistics

In this section, we discuss structure and arrangements in spreadsheets. We report on the dimensions of annotated tables. Specifically, we measure the height (#rows) to width (#columns) ratio, separately for single-table and multi-table sheets. The distributions are presented in Figure 3.i (outliers are omitted). We notice that the ratio is much smaller for multi-table sheets. Moreover, we observe wide tables (width>height) almost two times more often in multi-table sheets.

Figure 3.j reports on the arrangement of elements in the sheet. We performed this study twice: first only for tables (when multiple), and afterwards for all regions (i.e., including cells outside tables, such as Titles and Notes). In multi-table sheets, vertical (top-bottom) arrangements are the most prevalent. Nevertheless, we notice a high number of cases with horizontal (left-right), or mixed (both vertical and horizontal) arrangements. When it comes to all regions, mixed arrangements are almost as common as vertical ones.

Takeaway 8: Users prefer to arrange content vertically. Nevertheless, we frequently observe mixed arrangements.

These results bring forward the limitations of approaches doing layout inference at row granularity, such as [9], [12]. Clearly, these works would perform poorly, when cells of the same row exhibit different layout roles. In such cases, approaches that infer the layout for individual cells or regions of cells, such as [15], [20], are more suitable.

V. Conclusions

In this paper we present DECO, a dataset of annotated spreadsheets for layout inference and table recognition. Nevertheless, the dataset could serve also other niches of spreadsheet research. Unlike previous works, the files and their annotations are made publicly available. We provide tools to extract the annotations, and even to create new ones. Additionally, our annotation methodology is described in detail, going through each individual phase. Furthermore, we perform a thorough study of the annotated sheets, testing claims and assumption held by related work. Our study shows that there are still open questions, and that related work has overlook or oversimplified some challenges. Therefore, we summarized our findings in the form of takeways for future research works.

References

[1] F. Hermans and E. Murphy-Hill, “Enron’s spreadsheets and related emails: A dataset and analysis,” in *International Conference on Software Engineering-Volume 2*. IEEE Press, 2015, pp. 7–16.

[2] M. Fisher and G. Rothermel, “The euses spreadsheet corpus: a shared resource for supporting experimentation with spreadsheet dependability mechanisms,” in *SIGSOFT*, vol. 30, no. 4. ACM, 2005, pp. 1–5.

[3] T. Barik, K. Lubick, J. Smith, J. Slankas, and E. Murphy-Hill, “F use: a reproducible, extendable, internet-scale corpus of spreadsheets,” in *Working Conference on Mining Software Repositories*. IEEE Press, 2015, pp. 486–489.

[4] F. Hermans, M. Pinzger, and A. van Deursen, “Detecting and refactoring code smells in spreadsheet formulas,” *Empirical Software Engineering*, vol. 20, no. 2, pp. 549–575, 2015.

[5] T. Schmitz, D. Jannach, B. Hofer, P. W. Koch, K. Schekotihin, and F. Wotawa, “A decomposition-based approach to spreadsheet testing and debugging,” in *VL/HCC*. IEEE Computer Society, 2017, pp. 117–121.

[6] J. Cunha, J. P. Fernandes, J. Mendes, and J. Saraiva, “Mdsheet: A framework for model-driven spreadsheet engineering,” in *International Conference on Software Engineering*. IEEE Press, 2012, pp. 1395–1398.

[7] R. Abraham and M. Erwig, “Inferring templates from spreadsheets,” in *Proceedings of the 28th international conference on Software engineering*. ACM, 2006, pp. 182–191.

[8] F. Hermans, M. Pinzger, and A. Van Deursen, “Automatically extracting class diagrams from spreadsheets,” in *European Conference on Object-Oriented Programming*. Springer, 2010, pp. 52–75.

[9] Z. Chen and M. Cafarella, “Automatic web spreadsheet data extraction,” in *International Workshop on Semantic Search over the Web*. ACM, 2013, p. 1.

[10] ——, “Integrating spreadsheet data via accurate and low-effort extraction,” in *Proceedings of the 20th ACM SIGKDD*. ACM, 2014, pp. 1126–1135.

[11] Z. Chen, S. Dadiomov, R. Wesley, G. Xiao, D. Cory, M. Cafarella, and J. Mackinlay, “Spreadsheet property detection with rule-assisted active learning,” in *CIKM*. ACM, 2017, pp. 999–1008.

[12] M. D. Adelfio and H. Samet, “Schema extraction for tabular data on the web,” *Proceedings of the VLDB Endowment*, vol. 6, no. 6, pp. 421–432, 2013.

[13] J. Eberius, C. Werner, M. Thiele, K. Braunschweig, L. Dannecker, and W. Lehner, “Deexcelerator: A framework for extracting relational data from partially structured documents,” in *CIKM*. ACM, 2013, pp. 2477–2480.

[14] R. Abraham and M. Erwig, “Header and unit inference for spreadsheets through spatial analyses,” in *VL/HCC*. IEEE, 2004, pp. 165–172.

[15] E. Koci, M. Thiele, Ó. Romero Moral, and W. Lehner, “A machine learning approach for layout inference in spreadsheets,” in *IC3K: volume 1: KDIR*. SciTePress, 2016, pp. 77–88.

[16] X. Wang, “Tabular abstraction, editing, and formatting,” University of Waretloo, Waterloo, Ontaria, Canada, Tech. Rep., 1996.

[17] J. L. Fleiss, “Measuring nominal scale agreement among many raters.” *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[18] E. Koci, M. Thiele, W. Lehner, and O. Romero, “Table recognition in spreadsheets via a graph representation,” in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 139–144.

[19] M. Bendre, V. Venkataraman, X. Zhou, K. C.-C. Chang, and A. Parameswaran, “Scaling up to billions of cells with dataspread: Supporting large spreadsheets with databases,” Tech. Rep.

[20] A. O. Shigarov and A. A. Mikhailov, “Rule-based spreadsheet data transformation from arbitrary to relational tables,” *Information Systems*, vol. 71, pp. 123–136, 2017.