Robust Table Information Extraction from Sustainability Reports: A Time-Aware Hybrid Two-Step Approach

Hendrik Weichel[1,2], Jörg Schäfer[1], Martin Simon[1]
Frankfurt University of Applied Sciences[1]
University of Huddersfield[2]
Correspondence: hendrik.weichel@fra-uas.de

Abstract

The extraction of emissions-related information from annual reports has become increasingly important due to the Corporate Sustainability Reporting Directive (CSRD), which mandates greater transparency in sustainability reporting. As a result, information extraction (IE) methods must be robust, ensuring accurate retrieval while minimizing false values. While large language models (LLMs) offer potential for this task, their black-box nature and lack of specialization in table structures limit their robustness – an essential requirement in risk-averse domains. In this work, we present a two-step hybrid approach which optimizes both accuracy and robustness. More precisely, we combine a rule-based step for table IE with a regularized LLM-based step, both leveraging temporal prior knowledge. Our tests demonstrate the advantages of combining structured rules with LLMs. Furthermore, the modular design of our method allows for flexible adaptation to various IE tasks, making it a practical solution for industry applications while also serving as a scalable assistive tool for information extraction.

1 Introduction

Environmental, social, and governance (ESG) considerations have rapidly become central to corporate accountability and risk assessment. In the European Union, the Corporate Sustainability Reporting Directive (CSRD)[1] mandates that organizations disclose a variety of sustainability metrics in their annual or sustainability reports. While large public companies' data points are often available from data vendors, this is usually not the case for small and medium-sized enterprises (SMEs), whose reports frequently vary in format, presentation, and structure. At the same time, financial institutions, insurance companies, and other stakeholders increasingly require precise and reliable data, such as carbon emissions and other key indicators, to feed into quantitative risk models, in line with directives from bodies such as the European Banking Authority (EBA)[2].

Despite the growing volume of reported ESG data, extracting the relevant numerical values from heterogeneous documents remains a challenging task. In this work, we focus on the most common requirement of extracting numerical values from tabular structures. Many reports feature tables with inconsistent layouts, unstructured text, and varying terminologies, making standard IE methods prone to errors or heavy manual intervention. Furthermore, any inaccuracies in extracting emissions data or related metrics can lead to flawed risk assessments and regulatory non-compliance, underscoring the need for a highly robust, automated extraction pipeline.

To address these challenges, we propose a modular hybrid approach that regularizes LLM-based table IE by integrating domain expertise with temporal prior information. We demonstrate that combining rule-based techniques with machine learning models yields high accuracy, robustness, and scalability. Our table IE approach consists of two steps: A rule-based step that generates a candidate set containing the true information with high confidence and an LLM-based step that assists the user in selecting the most relevant element from this set. Our approach effectively addresses challenges such as mislabeled table headers, inconsistent data formats, and variations in corporate reporting styles. Most importantly, it reliably detects cases where the desired data point cannot be determined with confidence, ensuring transparency and trustworthiness in the extracted information. To the best of our knowledge, this is the first work to develop a table IE algorithm specifically tailored to the regulatory

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2464

[2] https://www.eba.europa.eu/publications-and-media/press-releases/eba-publishes-its-final-guidelines-management-esg-risks

Proceedings of the 2nd Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2025), pages 233–244
July 31, 2025 ©2025 Association for Computational Linguistics

requirements of financial institutions.

The remainder of this paper is organized as follows. Section 2 provides an overview of existing research related to table IE. Section 3 presents the proposed methodology. Section 4 describes our experimental setup and the datasets used to evaluate performance. Section 5 summarizes our empirical results and discusses the practical implications for stakeholders. Finally, Section 6 concludes the paper by highlighting the method's potential benefits and directions for future research. All our data is available on Github[3].

2 Related Work

The analysis of annual reports for climate-related information is an active area of research. Webersinke et al. (2022) introduce ClimateBert, a deep learning model based on BERT. In Bingler et al. (2024), it is applied to detect climate-related cheap talk in annual reports. In Schimanski et al. (2023), it is used to detect corporate, national, and regional net zero and reduction targets. The OS-Climate initiative, hosted by The Linux Foundation, recognized the need to extract key emission data from annual reports to facilitate climate-aligned financial decision-making. To address this, their Data Commons project (OS-Climate, 2025) offers an NLP toolkit for table data extraction. Mishra et al. (2024) explores table IE of ESG metrics. Their methodology translates tables into structured text using sequence-to-sequence transformer models. LLMs are also being explored for extracting financial data from tables in corporate reports. Balsiger et al. (2024) evaluates ChatGPT-4 and BARD for extracting key financial figures, such as balance sheets and income statements, from PDF-based annual reports. Their study highlights the potential and limitations of LLMs in processing complex financial tables. Wang et al. (2023) and Lamott et al. (2024) demonstrate that enriching prompts with OCR-derived layout information improves LLM document understanding; however, neither approach explicitly targets robustness in table extraction. Looking at the more technical research about table IE, the study by Lu et al. (2024) gives an overview of current research about table related tasks for transformer-based language models. Before the advent of large-scale LLMs (i.e., models with fewer than one billion parameters), researchers sought to enhance table understanding through architectural modifications, improved encoding methods, and model fine-tuning (Herzig et al., 2020; Iida et al., 2021; Deng et al.). With the emergence of LLMs, two strategies became dominant: fine-tuning and prompt engineering. The inputs typically include metadata along with the full table contents and a task-specific instruction. A more recent advancement in LLM-driven table extraction involves agent-based methods, which utilize LLMs' reasoning capabilities. Techniques such as Chain-of-Thought (CoT) prompting (Wei et al., 2023) and ReAct prompting (Yao et al., 2023) enable iterative extraction, refining the data retrieval process through step-by-step reasoning.

Despite these promising developments, a research gap remains in ensuring the robustness of these methods in risk-averse application domains. Purely LLM-based approaches inherently lack this robustness: On the one hand, their statistical nature limits reliability, and on the other, their inherently one-dimensional input representations conflict with the two-dimensional structure of tables. At the same time, academic literature highlights a disconnect between industry and academia. Chiticariu et al. (2013) state that "while rule-based IE dominates the commercial world, it is widely regarded as dead-end technology by academia." They observe, however, that rule-based methods remain essential in the industry. Unlike purely statistical machine learning approaches, rule-based systems leverage expert knowledge to define explicit patterns (e.g., regular expressions, ontology schemas, or grammar rules) that target relevant information. Rule-based table IE has been explored more extensively in other domains. For example, Potvin et al. (2016) propose a position-based rule-based method that utilizes the spatial arrangement of text elements to infer relationships.

3 Methodology

Let $\mathcal{R}$ denote a finite set of company annual and sustainability reports. Suppose we aim to extract a numerical value $y_t \in \mathbb{R}$, where t represents the year of the report. An example of such a value, which will serve as our running example, is "Scope 3 emissions in 2023 (in tonnes CO2 equivalents)" from the report r_{2023}. An IE algorithm provides a function $f\colon \mathcal{R} \to \mathbb{R}$, where $f(r_t)$ represents the best estimate of the true value y_t contained in the report r_t. Our approach integrates both domain ex-

[3] https://github.com/hendrikweichel/hybrid_2_step_table_information_extraction

pertise and temporal prior information, leveraging validated data from previous reports of the same company, i.e., $r_{t-1}, \dots, r_{t-n}$. Including such prior information into the IE pipeline can be interpreted as a regularization method, cf. Appendix A. Henceforth, the objective is to develop a reliable IE algorithm such that

$$f(r_t | r_{t-1}, \dots, r_{t-n}) = y_t \quad \forall r_t \in \tilde{\mathcal{R}} \subset \mathcal{R}$$

where we use the notation $f(\cdot | r_{t-1}, \dots, r_{t-n})$ to indicate the dependency of the function f on the parameters $r_{t-1}, \dots, r_{t-n}$, with $|\tilde{\mathcal{R}}|$ as large as possible, while ensuring that

$$f(r_t | r_{t-1}, \dots, r_{t-n}) = \infty \quad \forall r_t \in \mathcal{R} \setminus \tilde{\mathcal{R}}$$

to indicate cases where the function cannot reliably determine y_t. We base our approach on two key empirical observations made by domain experts analyzing sustainability reports:

(i) Emission data is almost always presented in tabular form.

(ii) Historical data provides a valuable prior for validating extracted values.

Thus, we assume that all emission values in $\mathcal{R}$ are stored in tables and define $\mathcal{T}(r_t) = \{\mathbf{T} | \mathbf{T} \in (\mathbb{R} \cup \Sigma)^{n \times m}; m, n \in \mathbb{N}\}$, where Σ represents textual values and $\mathbb{R}$ represents numerical values, as the set of all tables within report r_t. Our proposed IE function f is provided by a recursive approach, assuming that the relevant information has been successfully extracted and persisted in the previous years. In practice, the initial year is labeled manually. We follow a three-step pipeline:

1. **Table Extraction:** Given input r_t, extract the set of all tables $\mathcal{T}(r_t)$ using a table extraction method.

2. **Information Retrieval (IR):** Given the input r_t, ..., r_{t-n} (as well as the relevant tables and values extracted by the table extraction method in the previous years, see (iii) below), identify a table $\widehat{\mathbf{T}}(r_t) \in \mathcal{T}(r_t)$ that contains y_t (as well as y_{t-1} or even a longer history).

3. **Information Extraction (IE):** To extract the target value y_t, we apply a mapping

$$\widehat{\mathbf{T}}(r_t) \times \dots \times \widehat{\mathbf{T}}(r_{t-n}) \times \\ y_{t-1} \times \dots \times y_{t-n} \mapsto y_t.$$

$\widehat{\mathbf{T}}(r_{t-1})$, ..., $\widehat{\mathbf{T}}(r_{t-n})$ denotes the tables containing y_{t-1}, ..., y_{t-n} as extracted by the table extraction method in the previous years.

This paper focuses on step 3 of the pipeline, extracting information from tables, which is abbreviated as table IE in the following. While LLMs could, in principle, learn the complex mapping for table IE, $\widehat{\mathbf{T}}(r_t) \mapsto y_t$, there is one limiting factor making them unreliable for precise data extraction in regulatory settings: They are prone to hallucinations. This is further complicated by their inability to perceive the two-dimensional structure of tabular data due to their one-dimensional input format. To solve this problem, we present two distinct contributions:

1. A **rule-based table information extraction** approach to systematically extract y_t from $\widehat{\mathbf{T}}(r_{t-1})$. It exploits the historical knowledge about previous extractions and selects a candidate set of l possible solutions

$$\{\widehat{y}_t^{(1)}, \dots, \widehat{y}_t^{(l)}\} \in \widehat{\mathbf{T}}(r_t)$$

that has a high probability of uniquely containing y_t and a low probability of only returning candidates different from y_t. Applications that do not allow the use of LLMs, can apply this rule-based table IE like so:

$$f(r_t | r_{t-1}, \dots, r_{t-n}) = \begin{cases} \widehat{y}^{(1)}, & l = 1 \\ \text{None}, & \text{else} \end{cases}$$

2. A **hybrid two-step table information extraction** approach expands the rule-based table IE by leveraging the candidate set to regularize table IE with LLMs. We demonstrate in Section 3.2 below that the mapping

$$\widehat{\mathbf{T}}(r_t) \times \{\widehat{y}_t^{(1)}, \dots, \widehat{y}_t^{(l)}\} \times y_{t-1} \mapsto y_t$$

can be implemented through LLMs, both optimizing the robustness and accuracy of standard table IE through LLMs. We show that the rule-based pre-processing serves as a regularization mechanism for the LLM's table IE task. Still, given their black-box character, such a hybrid approach should assist in manual extraction rather than a fully automated solution in domains that require maximum robustness.

Note that our IE process is both recursive and highly modular, enhancing its flexibility and reliability. We extensively leverage this modularity

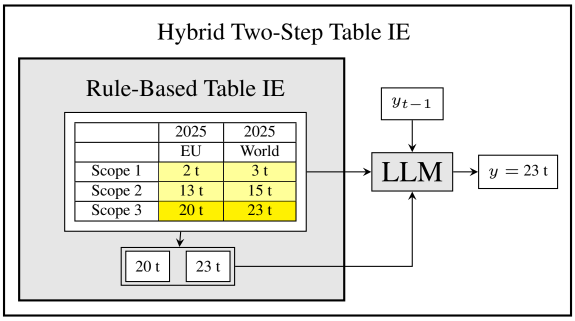

Figure 1: Illustration of our two contributions: (1) a rule-based table IE approach, and (2) a hybrid table IE method that builds upon (1) by leveraging its output.

Algorithm 1 Computation of scores for table cells

```
Require: $\mathbf{M}, \mathbf{O} \in \mathbb{R}^{n \times m}$
for k in $1, \dots, q$ do
    for $\widehat{\mathbf{T}}(r_t)_{i,j} \in \widehat{\mathbf{T}}(r_t)$ do
        $\mathbf{M}_{i,j} \leftarrow S^{(k)}(Q^{(k)}, \widehat{\mathbf{T}}(r_t)_{i,j})$
    end for
    $\mathbf{v} \leftarrow \max(\mathbf{M}, \text{dim} = d^{(k)})$
    $\mathbf{M}_{select} \leftarrow \text{tile}(\mathbf{v}, \text{shape} = \mathbf{M}(r_t).\text{shape})$
    $\mathbf{M}_{max} \leftarrow \mathbf{M}.\text{where}(\mathbf{M}_{i,j} = \max(\mathbf{M}))$
    $\mathbf{O} \leftarrow \mathbf{O} + \mathbf{M}_{select} - \mathbf{M}_{max}$
end for
```

to optimize our method for robustness, ensuring a low probability of incorrect outputs. Instead of returning erroneous results, the system is designed to return None when confidence is insufficient.

3.1 Rule-Based Table IE

Purely LLM-based table IE methods fail to utilize the two-dimensional nature of tables, cf. (Lu et al., 2024), and as a result, they overlook the implicit knowledge embedded within the matrix structure of tables $\widehat{\mathbf{T}}(r_t) \in (\mathbb{R} \cup \Sigma)^{n \times m}$. Our approach takes into account this knowledge by individually scoring all columns and rows based on their alignment with the target extraction. The different scoring methodologies tested in this work are presented in Section 3.1.1 below. Ultimately, cells that intersect in both the highest-scoring columns and rows are selected as the candidate set of values $\{\widehat{y}^{(1)}, \dots, \widehat{y}^{(l)}\} \subset \{\widehat{\mathbf{T}}(r_t)_{i,j} \mid i \in \{1, \dots, n\}, j \in \{1, \dots, m\}\}$. To ensure that the candidate set contains y_t with high confidence, we gather a set of constraints $\{C_1, \dots, C_q\}$ that apply to all the columns and rows that contain y_t. Such as, e.g., y_t always lies in a column which is annotated with the year t.

Algorithm 1 outlines the process of generating the candidate set, which is further illustrated in Figure 2. Based on the constraints, the algorithm assigns a score to each cell $\widehat{\mathbf{T}}(r_t)_{i,j}$ expressed in the score matrix $\mathbf{O} \in \mathbb{R}^{n \times m}$. Each constraint C_k, $k \in \{1, \dots, q\}$, is formalized as a triplet

$$C^{(k)} = (Q^{(k)}, S^{(k)}(c, Q), d^{(k)}),$$

where $Q^{(k)}$ is the query, e.g., the year of the searched emissions; $S^{(k)}(c, Q)$ is a similarity metric, that calculates the similarity score between a cell c and the respective query $Q^{(k)}$; and $d^{(k)}$ specifies the application orientation of $Q^{(k)}$ and $S^{(k)}$, indicating whether they are applied across rows or columns. These constraints encapsulate all prior knowledge about the target extraction that can be derived from $\widehat{\mathbf{T}}(r_{t-1}) \times \cdots \times \widehat{\mathbf{T}}(r_{t-n})$.

Besides the constraints for rows and columns, we apply additional constraints on the individual cell level. If the cell $\widehat{\mathbf{T}}(r_t)_{i,j}$ does not match the format of our target extraction, we set the corresponding score $\mathbf{O}_{i,j}$ to zero. In our example, where the goal is to retrieve numerical emission values, we exclude all cells that do not contain numbers or that include financial figures and percentages, as indicated by their corresponding units (€, \$, £, %).

As a final step, the cells of the table $\widehat{\mathbf{T}}(r_t)$ with the highest scores in $\mathbf{O}$ are selected as the candidate set:

$$\{\widehat{y}^{(1)}, \dots, \widehat{y}^{(l)}\} = \{\widehat{\mathbf{T}}(r_t)_{i,j} | \mathbf{O}_{i,j} = \max(\mathbf{O})\}$$

In production practice, an additional layer for identifying implausible results could be implemented by leveraging the time series of target values $y_t, \dots, y_{t-n}$. Candidate values $\widehat{y}^{(\cdot)}$ with a high deviation from the previous value y_{t-1} can be flagged as implausible. In practical terms, this involves calculating the difference between each candidate $\widehat{y}^{(\cdot)}$ and y_{t-1}, then flagging all candidates where $|\widehat{y}^{(\cdot)} - y_{t-1}|$ exceeds a predefined threshold.

The proposed modular and recursive design enables robust IE. More precisely, leveraging this modularity is essential for selecting robust similarity metrics and comprehensive constraint sets to accurately identify the row and column containing the target value y_t. As demonstrated in 4.2.1, where we conduct a cross-validation, this approach ultimately increases the likelihood of retrieving y_t as a candidate value.

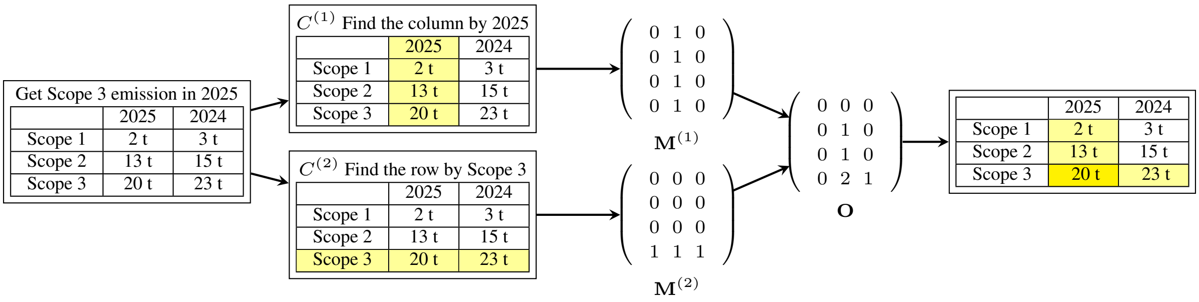

Figure 2: Flow chart of Algorithm 1.

3.1.1 Similarity Metrics

The similarity metrics take a query and a cell as input and assign a score between 0 and 1, reflecting the degree to which the cell matches the query, i.e.,

$$S(c, Q) \to [0, 1]$$

We perform cross-validation across several different metrics to determine the best-performing metrics for each query type; a comprehensive definition of all similarity metrics is given in Appendix B.

Regular expressions represent the simplest similarity metrics, applied either as exact string matching (*“is Q in c?”*) or in combination with pre-processing methods: We examine pre-processing through only selecting the numerical sub-strings of the query and the cell, and then carry out the string matching. Furthermore, one can tokenize the query into subqueries and compute the share of subqueries that are contained in the cell. This leaves more degrees of freedom for the structure of the cell strings and enables the use of continuous scores between 0 and 1. For the same reason, we examine the set-based **Jaccard similarity** and the **Levenshtein distance**. Both, in theory, accept minor dissimilarities between cell and query and could lead to a higher precision.

Additionally, we evaluate **semantic vector-based matching**. Techniques such as Word2Vec (Mikolov et al., 2013) and transformer-based word embedding models[4] have demonstrated strong performance in measuring similarity. These models assign vectors to sentences, enabling similarity measurement based on the comparison of their vector embeddings. A drawback of these machine learning models is their black-box nature and higher computational cost compared to the previously discussed methods.

Numerical metrics can be used to compare cells and queries containing numerical content. To do so, both the cell and the query are converted to floats. We test both, a numerical metric that returns the percent-wise deviation from the cell value to the query value and one binary numerical metric that returns 1 if the absolute difference is smaller than a given threshold and 0 else.

In Section 4, we perform cross-validation to determine which similarity metric fits best for which type of query. Appendix B formally defines all the queries we tested.

3.2 Hybrid Two-Step Table IE

If the rule-based table IE does not return a unique candidate, its candidate set can be used to assist table IE with LLMs. A straightforward table IE task would instruct the LLM to return the target value y_t, given the table $\widehat{\mathbf{T}}(r_t)$. We shift this question and answer task to a regularized binary classification task: Given the table $\widehat{\mathbf{T}}(r_t)$ and the previous year's emission y_{t-1} (if contained in $\widehat{\mathbf{T}}(r_t)$), we instruct the LLM to select y_t from $\{\widehat{y}^{(1)}, \dots, \widehat{y}^{(l)}\}$. Note that this approach offers a two-fold regularization of the problem: first, by incorporating prior information (cf. Appendix A), and second, by constraining the solution space. This enhances the robustness of table IE using LLMs. Our instruction prompt is structured as follows:

Table IE by Selection Prompt

Context: $\widehat{\mathbf{T}}(r_t)$
Instruction: Choose the element from the list of candidate lists that contains the total Scope 3 emissions in the year t given in the table $\widehat{\mathbf{T}}(r_t)$ in JSON format. The previous year's emissions were y_{t-1}, and it is likely that this year's emissions do not deviate significantly from y_{t-1}.
Candidate list: $\{\widehat{y}^{(1)}, \dots, \widehat{y}^{(l)}\}$

[4] We use the models from Song et al. (2020) and Wang et al. (2020), with fine-tuning in https://huggingface.co/sentence-transformers/all-mpnet-base-v2 and https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2, respectively.

4 Experiments

This section presents our experiments on testing the table extraction approach using our running example of extracting Scope 3 emissions from financial institutions' annual reports. In these experiments, we used the following queries: (1) Filter the columns by the emission year t; (2) Filter rows by the emission type *"Scope 3"*; (3) Filter rows by the previous year's emission y_{t-1}; (4) Given the fact that the table structure frequently remains unchanged, with consistent row and column descriptions, we leverage this stability and use the name of the row in $\widehat{\mathbf{T}}(r_{t-1})$ that contains y_{t-1} as a query to filter the rows in $\widehat{\mathbf{T}}(r_t)$. In some cases, the first column can be None; we then take the first cell in the row that contains a textual value.

We test the purely rule-based table IE in two steps: First, we cross-validate several similarity metrics for each of the used query types to identify the robust metrics. Second, we choose the robust similarity metrics and combine them to test the creation of a candidate set. Here, we aim to validate that y_t is identified with high probability within the candidate set. We evaluate both the rule-based table IE and the hybrid two-step table IE approach against a benchmark – a straightforward LLM-based extraction.

4.1 Dataset

We test our approach by extracting Scope 3 greenhouse gas emissions from tables in the annual reports of Europe's largest banks. This represents a particularly relevant real-world scenario, as Scope 3 emissions constitute the most significant emission category for financial institutions, given that they encompass financed emissions. At the same time, Scope 3 emissions are notoriously difficult to quantify, often resulting in frequent restatements from year to year, thus providing an ideal testbed for our table IE approach. Note that this table IE methodology should, in a subsequent step, be integrated into a full IE pipeline, as outlined in Section 3. Since step two of this pipeline, Information Retrieval, ensures that the retrieved tables contain the emission, our dataset consists exclusively of reports including tables that contain the Scope 3 emissions.

For calibration and testing, we retrieved the 52 largest European banks by market capitalization and examined their annual reports between 2018 and 2023. The Scope 3 emissions were initially extracted manually from each report and tagged with their corresponding page numbers. These values represent the extraction target y_t. Using an AWS-based OCR system (see (EdenAI, 2025)), we extracted a set of candidate tables from the page that contains y_t. We then automatically selected only the table $\widehat{\mathbf{T}}(r_t)$ that contains y_t. Subsequently, we ensured that the structured tables accurately preserved the original formatting and structure as presented in the PDF versions of the annual reports. Any deviations from the original table structure were corrected manually, because the final pipeline must preserve layout fidelity while discarding only those tables that lack the target value y_t. Automatic detection of deviations will be explored in future work as part of the Information Retrieval step. The rule set was calibrated on a separate dataset drawn from a distinct group of banks.

4.2 Rule-Based Table IE

To evaluate the rule-based table IE, we adapt the notion of a binary classification that classifies each cell in the table $y \in \widehat{\mathbf{T}}(r_t)$ into one of two classes:

1. *positive*: $y \in \widehat{\mathbf{T}}(r_t)$ is a candidate for y_t due to the structure of $\widehat{\mathbf{T}}(r_t)$, these are all the candidates $\{\widehat{y}_t^{(1)}, \ldots, \widehat{y}_t^{(l)}\}$.

2. *negative*: $y \in \widehat{\mathbf{T}}(r_t)$ is not a candidate for y_t due to the structure of $\widehat{\mathbf{T}}(r_t)$, these are all the elements in the complement set $\{y|\ y \in \widehat{\mathbf{T}}(r_t)\} \setminus \{\widehat{y}_t^{(1)}, \ldots, \widehat{y}_t^{(l)}\}$.

That is, the predicted *positives* are the candidates, and the predicted *negatives* are all other elements in $\widehat{\mathbf{T}}(r_t)$. The *true* value is the extraction target y_t, and the *false* values are all other elements. This type of table IE is considered robust if it consistently includes y_t in the candidate set. Naturally, this may come at the cost of retrieving more false positives, resulting in a larger candidate set. In terms of the classification problem, our goal is to minimize false negatives and optimize recall. For example, if the candidate set contains the only element for y_t, the recall is 100.00%. Naturally, this introduces a recall-precision trade-off: including all elements $\{y|\ y \in \widehat{\mathbf{T}}(r_t)\}$ in the candidate set would result in a recall of 100% but a significantly lower precision score. A precision of 100.00% would occur, for instance, if the sole candidate $\widehat{y}^{(1)}$ is y_t. We additionally use the notion of *false positives only (FPO)*, which describes the share of extractions where only false positives were returned.

	Find column that contains y_t with query t			Find row that contains y_t with query y_{t-1}		
	recall	prec.	FPO	recall	prec.	FPO
Regex						
Complete	97.62	78.97	**0.00**	38.10	38.10	**0.00**
Numerical	**100.00**	78.97	**0.00**	38.10	38.10	**0.00**
Word Wise	97.62	78.97	**0.00**	45.24	40.66	**0.00**
Numerical Metrics						
Binary	80.95	71.03	**0.00**	40.48	40.48	**0.00**
Continuous	88.10	**79.76**	11.90	64.29	**64.29**	35.71
Step	95.24	76.97	4.76	**71.43**	53.97	28.57

Table 1: Test performance of numerical similarity metrics for the numerical queries to find the required rows and columns (cf. Section 4.2.1).

Given one particular table, FPO is 1, if a nonempty candidate set disjoint from $\{y_t\}$ is returned, and 0 otherwise. Recall, precision, and FPO report the average values across all extractions in the dataset.

4.2.1 Similarity Metrics Cross-Validation

The similarity metrics are used to find those rows or columns in $\widehat{\mathbf{T}}(r_t)$ that contain the extraction target y_t. In our running example, we use four different queries to do this. The cross-validation provided here evaluates a selection of similarity metrics (see Section 3.1.1 and Appendix B) with respect to their ability to individually identify the rows or columns in $\widehat{\mathbf{T}}(r_t)$ that contain y_t. Queries are classified into two categories: numerical and textual. Table 1 presents the results for **numerical queries**, specifically the year t and the previous year's emissions y_{t-1}; we apply numerical metrics and regular expressions. Table 2 presents the results for **textual queries**, including the emission type and the row name of the row in $\widehat{\mathbf{T}}(r_{t-1})$ that contains y_{t-1}, we apply several NLP similarity metrics such as simple regular expressions, Levenshtein distance, Jaccard similarity, and embedding-based similarities.

	Find row that contains y_t with emission type			Find row that contains y_t with prev. table's row name		
	recall	prec.	FPO	recall	prec.	FPO
Regex						
Complete	**100.00**	**89.84**	**0.00**	64.29	60.71	**0.00**
Word Wise	**100.00**	77.94	**0.00**	**100.00**	80.38	**0.00**
Levenshtein	57.14	45.36	42.86	83.33	79.76	16.67
Jaccard Similarity						
4-grams	71.43	69.05	28.57	88.10	84.52	11.90
5-grams	73.81	73.81	26.19	88.10	84.52	11.90
6-grams	80.95	80.95	19.05	90.48	**86.90**	9.52
7-grams	85.71	85.71	14.29	88.10	84.52	11.90
Embedding						
All MiniLM	59.52	59.52	40.48	85.71	82.14	14.29
MPNet Base	54.76	54.76	45.24	85.71	82.14	14.29
Word2Vec	40.48	40.48	59.52	88.10	84.52	11.90

Table 2: Test performance of textual similarity metrics for the textual queries (cf. Section 4.2.1).

4.2.2 Test Rule-based Table IE

To evaluate the proposed table IE approaches, we define the following constraint set, obtained from the most robust similarity metrics in the cross-validation, i.e.,

1. (t, Reg. Ex. Numerical, column)
2. (y_{t-1}, Numerical Binary, row),
3. ("Scope 3", Reg. Ex. Complete Strings, row),
4. (x_{t-1} row name, Reg. Ex. String-Level, row)

The average recall of the table IE experiments with this set of constraints was 100%, the average precision was 89.65% and the extraction uniquely identified y_t as the sole element in the candidate set in 80.95% of all extractions.

4.3 Hybrid Two-Step Table IE

Testing the full table IE, i.e., retrieving a single candidate for y_t rather than a set of candidates, involves a slightly different notion of false positives and false negatives than we used for the test of the rule-based table IE, since the result is no longer a set of candidates but either a single value for y_t or None. In this context, a *true positive extraction* is selecting the correct element y_t, selecting a candidate different from y_t is considered a *false positive*. A *false negative* when None was returned despite $\widehat{\mathbf{T}}(r_t)$ containing y_t. Analogously, *true negative* occurs when y_t is not contained in $\widehat{\mathbf{T}}(r_t)$ and None is correctly extracted. It is crucial to emphasize that, unlike the rule-based table IE in the first step, which focuses on minimizing false negatives when creating a candidate set, a robust second step that selects only one element prioritizes minimizing false positives, thereby optimizing precision. Table 3 presents benchmark results for a straightforward LLM-based table IE.

LLM	recall	prec.
GPT-4o	95.23	100.00
GPT-4o-mini	93.65	100.00
Deepseek r-1	90.91	95.65
llama 70b	90.48	100.0
llama 8b	86.11	83.78

Table 3: Benchmark for extracting y_t from $\widehat{\mathbf{T}}(r_t)$ with straightforward Table IE by LLMs.

Our methodology yielded the following results on the same dataset:

- The **rule-based table IE** achieved a **precision of 100%**, meaning that it never extracted an incorrect value for y_t. It also achieved a **recall of 80.95%**, indicating that in 80.95% of cases, the correct value y_t was extracted directly, while in the remaining 19.05% of cases, y_t was included in the candidate set.

- Our **hybrid two-step table IE** approach improved these results by utilizing an LLM to identify y_t within the candidate set generated by the rule-based method. For all LLMs listed in Table 3, i.e., GPT-4o, GPT-4o-mini, Deepseek r-1, llama 70b, and llama 8b, this approach successfully identified y_t, achieving both **precision and recall of 100%**.

5 Discussion

The cross-validation described in Section 4.2.1 enabled selecting the most robust similarity metrics, cf. Tables 1 and 2. Using regular expressions on numerical substrings is the most effective approach for identifying the column containing y_t, given the year t. It always identifies the right column and has a relatively high precision. We can also observe that identifying the row containing y_t given y_{t-1} works robustly using regular expressions and binary numerical metrics. Specifically, if y_{t-1} is present in $\widehat{\mathbf{T}}(r_t)$, our rule-based approach successfully detects it; otherwise, it correctly determines its absence. The fact that the latter case is observed rather frequently is not particularly surprising, given the fact that in our dataset companies' yearly Scope 3 emission restatements have a frequency of roughly 60%. However, through the EU's efforts to standardize sustainability reporting, it is likely that the frequency of restatements will decrease in the future. Table 2 presents the evaluation of textual metrics. The results indicate that identifying the correct row using textual metrics is highly robust when employing simple regular expressions. These methods consistently achieved a false positive only rate of 0% for both queries. However, finding the row based on the row name of y_{t-1} in the previous table did not achieve a 100% recall, suggesting that only 64.29% of row names remained unchanged from year to year. This issue is effectively addressed by word-level matching, which improves both precision and recall. The Levenshtein ratio and Jaccard similarity performed poorly, primarily because these metrics penalize differences in query and cell lengths, even when such variations do not affect the semantic meaning. Similarly, embedding-based similarities struggled because they treat numerically similar terms (e.g., "Scope 2" vs. "Scope 3") as nearly identical, leading to underperformance compared to simpler rule-based methods. In future work, we aim to explore how embedding-based similarities can be better adapted to improve performance. As a result of the cross-validation, we selected the four most robust similarity metrics and combined them to perform the rule-based table IE described in Section 4.2.2. We see that the recall is 100%, which means that the candidate set always contains the extraction target y_t. The proportion of extractions in our tests where the candidate set contained only y_t, i.e., exclusively returning true positives, is 80.95% for the most robust set of constraints. The average number of candidate values was below 1.5 for all sets of constraints. In summary, these results demonstrate that a robust IE can be ensured if the similarity metrics provide a consistently robust extraction. The tests described in Section 4.3 compared both steps of our table IE approach against a straightforward LLM-based approach. We observe that, in a table question-and-answer setting, only the models GPT-4o, LLaMA 70B, and GPT-4o-mini achieved a precision of 100%. In contrast, our hybrid two-step approach successfully performed information extraction with both recall and precision at 100%. These results demonstrate immense substantial gains in information extraction, especially for smaller LLMs such as llama 8B, thus highlighting the effectiveness of our regularization approach. Consequently, our approach enables the utilization of smaller, more cost-effective, and open-source models, enhancing accessibility and scalability. This factor is especially critical in the financial industry, which prefers open-source on-premise solutions and demands scalability.

6 Conclusion

In this paper, we presented a two-step hybrid table IE approach with a focus on robustness, making it well-suited for risk-averse application domains. As outlined in the problem statement, relying solely on an LLM is not feasible in such domains – an essential argument in favor of our approach. Additionally, candidate sets generated by our method include the extraction target with high probability, which can be leveraged to support manual data quality control and validation. We anticipate that evolving regulations for sustainability reporting will lead to higher data quality, greater consistency, and increased standardization. These trends further strengthen the effectiveness and applicability of the methodology presented in this paper.

Limitations

Our approach ensures robustness through customize the constraints of the extraction in a highly modular system. This is an advantage, however, it is important to exploit this customizability for other extraction tasks, i.e., it is important to specify queries and similarity metrics for other applications and / or other domains. A further limitation is that we tested the approaches for a rather small dataset and only used tables in a well-structured format. In future work, we plan to address these limitations.

Ethics Statement

No ethical concerns arise from the study, and all methodologies adhere to standard academic and scientific integrity principles. Additionally, no conflicts of interest are present, and the work complies with ethical guidelines for responsible research and publication.

Acknowledgements

The authors would like to express their gratitude to Ann Smith and George Bargiannis at University of Huddersfield, whose support and feedback significantly contributed to the development of this study.

Funding

This project (HA project no. 1647/23-200) is financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence). Martin Simon would like to acknowledge support by the German Federal Ministry of Education and Research (BMBF) under Grant No. 03FHP191. The study used computing infrastructure sponsored by the Bundesministerium fur Bildung und Forschung (BMBF)/German Federal Ministry of Education and Research, Funding program Forschung an Fachhochschulen, KI@FRA-UAS.

References

David Balsiger, Hans-Rudolf Dimmler, Samuel Egger-Horstmann, and Thomas Hanne. 2024. Assessing large language models used for extracting table information from annual financial reports. *Computers*, 13(10).

Julia Anna Bingler, Mathias Kraus, Markus Leippold, and Nicolas Webersinke. 2024. How cheap talk in climate disclosures relates to climate initiatives, corporate emissions, and reputation risk. *Journal of Banking & Finance*, 164:107191.

Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA. Association for Computational Linguistics.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. TURL: Table Understanding through Representation Learning.

EdenAI. 2025. Ocr table parsing apis - eden ai. `https://www.edenai.co/feature/ocr-table-parsing-apis`. Accessed: 28-January-2025.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456, Online. Association for Computational Linguistics.

Marcel Lamott, Yves-Noel Weweler, Adrian Ulges, Faisal Shafait, Dirk Krechel, and Darko Obradovic. 2024. Lapdoc: Layout-aware prompting for documents.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2024. Large Language Model for Table Processing: A Survey.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Lokesh Mishra, Sohayl Dhibi, Yusik Kim, Cesar Berrospi Ramis, Shubham Gupta, Michele Dolfi, and Peter Staar. 2024. Statements: Universal information extraction from tables with large language models for esg kpis. In *Proceedings of the 1st Workshop on Natural Language Processing Meets Climate Change (ClimateNLP 2024)*, page 193–214. Association for Computational Linguistics.

OS-Climate. 2025. Data commons. `https://os-climate.org/data-commons/`. Accessed: 2025-03-01.

Benoit Potvin, Roger Villemaire, and Ngoc-Tan Le. 2016. A Position-Based Method for the Extraction of Financial Information in PDF Documents. In *Proceedings of the 21st Australasian Document Computing Symposium*, pages 9–16. ACM.

Tobias Schimanski, Julia Bingler, Mathias Kraus, Camilla Hyslop, and Markus Leippold. 2023. ClimateBERT-NetZero: Detecting and assessing net zero and reduction targets. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15745–15756, Singapore. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.

Wenjin Wang, Yunhao Li, Yixin Ou, and Yin Zhang. 2023. Layout and task aware instruction prompt for zero-shot document image question answering.

Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. 2022. Climatebert: A pretrained language model for climate-related text.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models.

A Including prior information as a regularization method

Let us demonstrate that for IE purely driven by an LLM, including temporal prior information may be interpreted as a regularization method in a strict mathematical sense. While the method we propose here is a hybrid method rather than purely driven by an LLM, this may still serve as a motivation for including prior knowledge to obtain more robust methods. Xie et al. (2022) study in-context learning for LLMs trained on a pretraining distribution given by a HMMM. They prove that, under this assumption, the LLM implicitely performs Bayesian inference. We define the sequence of training examples $\mathcal{S}_n = (S_1, ..., S_n)$ such as "Scope 1 emissions in 2021 were ?? t", "Scope 1 emissions in 2020 were ?? t", and the test prompt x_{test} ="Provide the Scope 1 emissions in the year 2023 in the unit t".

The first step in in our framework provides an additional chunk $\mathcal{C}$ of text from the text corpus r_t which is appended to the training examples to obtain

$$\widetilde{\mathcal{S}}_n = (\mathcal{S}_n, \mathcal{C}).$$

Therefore, Equation 5 in Xie et al. (2022) becomes

$$\begin{aligned} p\big(y \mid \widetilde{\mathcal{S}}_n,\ x_{\text{test}}\big) \ \propto\ & \int_\theta \sum_{h \in \mathcal{H}} p\big(y \mid x_{\text{test}},\ h,\ \theta\big) \\ & \times p\big(\widetilde{\mathcal{S}}_n, x_{\text{test}} \mid \theta\big) \\ & \times p\big(h \mid \widetilde{\mathcal{S}}_n, x_{\text{test}}, \theta\big)\ p(\theta)\ \mathrm{d}\theta. \end{aligned}$$

In this setting, the prior $p(\theta)$ encodes the LLM's pretrained distribution. Including $\widehat{\mathcal{S}}$ in addition to the test prompt updates the model's posterior by answering the question which parts of the parameter space and which hidden states $h \in \mathcal{H}$ are most relevant with regard to the inputs, thus preventing the model from drifting to irrelevant states or modes.

B Similarity Metrics

Each similarity metric has inputs query and cell and returns a value between 0 and 1.

B.1 Regular Expression

To evaluate whether a query is contained within a string, we implemented five complementary approaches.

B.1.1 Complete Word Matching

The first approach converts both the query and cell string to lowercase and checks if the query is contained within the cell. It returns 1 for a match and 0 for no match.

B.1.2 Numerical Substring Matching

The second method extracts the numeric characters from both the query and cell string, then checks if the query's numbers appear in the cell. It returns 1 for a match and 0 for no match.

B.1.3 Word-Level Matching

The fourth method splits the query into words, converts them to lowercase, and calculates the fraction of words found in the target string. The result ranges from 0 (no matches) to 1 (all words matched).

B.2 Levenshtein Ratio

This method is based on the Levenshtein distance, which quantifies the minimum number of single-character edits – insertions, deletions, or substitutions – required to transform one string into another. To normalize this distance, the Levenshtein ratio divides it by the maximum possible string length, yielding a similarity score between 0 and 1, where 1 indicates identical strings and 0 denotes no similarity. This approach is particularly useful for handling minor spelling variations, typos, and fuzzy matching, making it a robust technique for evaluating approximate string containment.

B.3 Jaccard Similarity

The Jaccard-Coefficient is a statistic used for the similarity of two sets. In NLP this statistic is used to yield a similarity score between 0 and 1. During preprocessing, we create separate both the query and the cell into the sets A and B of strings with length n. Then we calculate the Jaccard similarity as so:

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

B.4 Numerical Comparison

Here, we preprocess the strings such that we obtain the quantity and unit. (*"20,000 t CO2" → Quantity: 20000, Unit: "t CO2"*). If there is no unit, we directly compare the two quantities. To create a similarity score between the two quantities a and b, we use the following methods.

B.4.1 Binary Comparison

The binary comparison returns the score 1 if the absolute difference between a and b is smaller than 1. Else it returns 0.

$$S(a,b) = \begin{cases} 1, & |a-b| < 1 \\ 0, & \text{else} \end{cases}$$

B.4.2 Continuous Comparison

To allow minor differences between a and b we use a continuous function. It returns the relative difference between with respect to a.

$$S(a,b) = max\left(\frac{|a-b|}{a}, 0\right)$$

B.4.3 Step Function

The step function is an extension of the continuous function.

$$S(a,b) = \begin{cases} 0, & |a-b| < 1 \\ 0.9, & \frac{|a-b|}{a} < 0.1 \\ 0.8, & \frac{|a-b|}{a} < 0.2 \\ 0.6, & \frac{|a-b|}{a} < 0.4 \\ 0.4, & \frac{|a-b|}{a} < 0.6 \\ 0.2, & \frac{|a-b|}{a} < 0.8 \\ 1.0, & \text{else} \end{cases}$$

B.4.4 Word Embeddings and Cosine Similarity

To calculate the similarities between two words a and b, we first generate the word embeddings with the given model $\mathbf{e}_a$ and $\mathbf{e_b}$. Then we define the similarity of a and b as:

$$S(a,b) = \frac{\mathbf{e}_a \cdot \mathbf{e}_b}{\|\mathbf{e}_a\|\|\mathbf{e}_b\|},$$

where $e_a \cdot e_b$ denotes the scalar product.

C Measurements for Experiments

These are the formal definitions for the experiments to test the rule-based approach to create a candidate set:

$$\text{Average Precision} = \frac{1}{N}\sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i}$$

$$\text{Average Recall} = \frac{1}{N}\sum_{i=1}^{N} \frac{TP_i}{TP_i + FN_i}$$

$$\text{FPO} = \sum_{i=1}^{N} \mathbb{1}(TP_i = 0 \wedge FN_i = 0 \wedge FP_i > 0),$$

where N is the total number of test extractions, and TP_i, FP_i, FN_i correspond to the counts for extraction i.

These are the formal definitions for the experiments to test the full table IE:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

D Prompting

We used the following prompt for our benchmark of table IE:

Benchmark Table IE Prompt

System: Help to extract the total Scope 3 emissions in the year t from a table given below.
Human: Therefore, choose the best answer for the given context. And fill in the json format: "Scope 3": <Scope 3 emissions>, where <Scope 3 emissions> is a string of the Scope 3 emission with unit.
Context: $\widehat{\mathbf{T}}(r_t)$
Question: What are the total scope 3 emissions in the year t given in the table?

We used the following prompt for our hybrid two-step table IE:

Table IE with candidate set

System: Help to extract the total Scope 3 emissions in the year t from a table given below.
Human: Help to extract the total Scope 3 emissions in the year t from a table given below from a preselection of possible answers. The previous year's emissions were y_{t-1}, and it is likely that this year's emissions do not deviate significantly from y_{t-1}. Therefore, choose the best answer for the given context out of the set of possible answers. And fill in the json format: {"Scope 3": <Scope 3 Emission>}, where <Scope 3 Emission> is a string of the Scope 3 Emission with unit.
Context: $\widehat{\mathbf{T}}(r_t)$
Question: What are the total Scope 3 emissions in the year t given in the table?
Select one of these possible answers $\{\widehat{y}^{(1)}, \dots, \widehat{y}^{(l)}\}$ and make sure that it keeps the JSON format.