Text2Tabular – Reconstructing Tabular Research Data from Scientific Publications

Jonas Gottal and Florian Matthes
Technical University of Munich
School of Computation, Information and Technology
Department of Computer Science
{jonas.gottal, matthes}@tum.de

Abstract

The increasing reliance on data-driven research underscores the need for accessible datasets, particularly in the medical domain. However, raw datasets are frequently unavailable due to privacy constraints and ethical considerations, which complicates reproducibility, meta-analyses, and large-scale data-driven research. Text2Tabular addresses this challenge by reconstructing research datasets from scientific publications using advanced natural language processing and statistical modeling. Our key contributions include: (1) a unified framework combining Large Language Model driven information extraction with copula-based distribution modeling, (2) novel integration of statistical test results as distribution constraints through constrained Markov Chain Monte Carlo refinement, and (3) an own comprehensive benchmark comprising real scientific publications with corresponding raw datasets for evaluating our literature-based data reconstruction. Evaluation on both benchmark datasets and our curated collection demonstrates strong performance in Train-on-Synthetic-Test-on-Real (TSTR) evaluations, alongside accurate replication of descriptive statistics showing that Text2Tabular preserves the statistical properties and multivariate relationships of the original datasets. Text2Tabular facilitates scientific progress by enabling immediate access to realistic, domain-specific synthetic data, thus, improving data accessibility, and mitigating data scarcity in fields with limited real-world data.

1 Introduction

Access to high-quality, well-reported datasets remains a key barrier to reproducibility and large-scale secondary research, especially in privacy-sensitive domains. Raw tabular data often remains inaccessible, while study results are routinely available only as statistical summaries, group-level measures, or test outcomes.

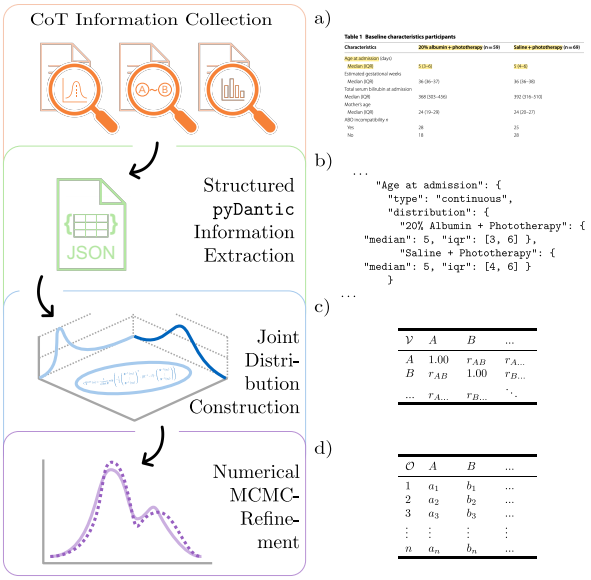

Figure 1: Overview of the Text2Tabular pipeline with examples: (a) extraction of relevant information from the publications, including marginal statistics, correlations, and statistical test results (Magai et al., 2019b, p. 3); (b) structuring extracted data; (c) analytical construction of a joint distribution using copula functions; and (d) numerical optimization via MCMC sampling.

This disconnect limits meta-analyses, reproducibility assessments, and the application of modern machine learning methods that require large amounts of well-structured raw sample-level data. Text2Tabular closes this gap by automatically reconstructing synthetic datasets from scientific literature. Our approach combines Natural Language Processing (NLP) for structured information extraction with a copula-based statistical engine that builds synthetic multivariate data, constrained by the published summary statistics and key test results. Text2Tabular accelerates scientific progress by providing immediate access to realistic, domain-specific synthetic data while maintaining privacy.

Figure 1 illustrates the workflow: marginal statistics, correlations, and statistical test outcomes are parsed, structured, then synthesized into realistic

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6682–6697

sample-level data. This enables analyses that would otherwise require inaccessible raw datasets.

$\mathcal{V}$	Group 1	Group 2	Total
A	n_{A1}	n_{A2}	n_A
B	$\bar{B}_1 \pm \sigma_{B1}$	$\bar{B}_2 \pm \sigma_{B2}$	$\bar{B} \pm \sigma_B$
C	n_{C1}	n_{C2}	n_C

Figure 2: Example of a data summary table for variables $\mathcal{V}$ from a randomized controlled trial (RCT) with counts n, means $\bar{V}$ and standard deviations σ.

Most empirical studies present data summaries (e.g., Figure 2) – such as contingency tables with means and standard deviations – alongside statistical test results embedded within the text. Our reconstruction method accommodates diverse reporting formats, focusing on the more complex RCTs and observational studies, by incorporating relationships captured through marginal summaries and inferential tests. To guide data reconstruction, we concentrate on extracting the most common test statistics (see Figure 3 for an overview of widely used tests in empirical publications (Armitage et al., 2013; Mishra et al., 2019; Yan et al., 2017)), as well as group structures and key statistical measures – including central tendency (mean, median, mode), dispersion (standard deviation, variance, IQR), range, and percentiles (quantiles, confidence intervals). These extracted parameters form the backbone of the synthesized joint distribution and facilitate the reconstruction of ordinal, continuous, categorical and binary data.

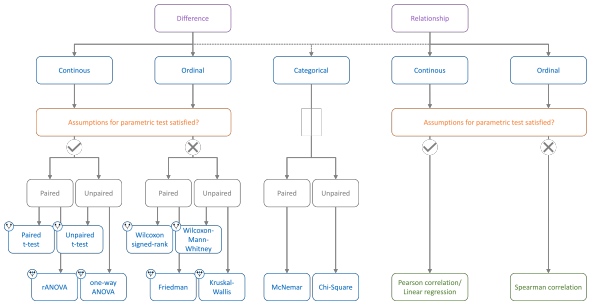

Figure 3: Overview of the statistical tests used in empirical scientific publications. (Til, 2022)

This work, as a feasibility study, addresses the most common scenarios and measure types encountered in practice (Feng et al., 2022). While not all possible statistical tests are currently implemented, the framework is designed for extensibility to broader data types such as time series and longitudinal cohorts in future iterations. Upon publication, the framework will be made available as an open-source Python package, enabling researchers to easily apply it to their own datasets and publications.

We make our code and data available in a public GitHub repository. [1]

2 Related Work

Recent progress in synthetic data generation has centered on transforming existing tabular datasets using generative models, including approaches based on probabilistic modeling (Shi et al., 2025; Patki et al., 2016; Nakamura-Sakai et al., 2024; Nguyen et al., 2024) and, more recently, transformer architectures (Hollmann et al., 2023, 2025). While highly effective at producing new samples, these frameworks assume direct access to structured input data and are not designed for scenarios where only unstructured scientific descriptions are available.

Complementary efforts in scientific information extraction rely on NLP to identify and extract statistical measures from publications (Polak and Morgan, 2024). However, such methods stop at information retrieval and do not attempt to synthesize new datasets or enable downstream statistical or machine learning analyses.

Approaches aimed at systematically scraping data from scientific literature (Pradhan et al., 2019) face persistent challenges: underlying data are often inaccessible, inconsistently formatted, or of insufficient quality for reliable synthesis – as also observed in our own evaluation.

Techniques for reconstructing data from statistical summaries (Heathers et al., 2018; Duchscherer et al., 2019; Jävergård et al., 2024) typically reconstruct only marginal distributions and ignore joint dependencies among variables. To date, no automated system exists that extracts structured information from publications and creates synthetic tabular datasets faithfully preserving both marginal and joint statistical properties. Crucially, reproducing the exact original data is inherently impossible from summary statistics alone; our aim is instead to generate domain-specific synthetic data that matches reported properties for immediate scientific utility.

In summary, three central limitations characterize prior work: (1) a strict separation between information extraction and data synthesis, (2) limited

[1] https://github.com/jonasgottal/Text2Tabular

preservation of complex joint structures, and (3) inadequate use of statistical test results as formal constraints on data generation. Text2Tabular closes these gaps, introducing an integrated pipeline that combines NLP-based extraction with copula modeling and constrained MCMC, ensuring generated data adhere to both reported marginals and multivariate relationships.

3 Foundations

We present the mathematical foundations of Text2Tabular with a focus on analytical, scalable methods for data reconstruction. Analytical solutions are prioritized to ensure computational efficiency at scale.

Definition 1 (Copula) *A copula is a d-dimensional cumulative distribution function with uniform marginals (Haugh, 2016; Nelsen, 2006).*

$$C(u_1, \ldots, u_d) = P(U_1 \leq u_1, \ldots, U_d \leq u_d), \tag{1}$$

where $U_i \sim$ Uniform$[0, 1]$ for $i = 1, \ldots, d$.

Copulas provide a flexible means to model dependencies between variables, allowing the joint distribution to be constructed from any set of marginals. This separation of marginals and dependence structure is essential for reconstructing multivariate data when only summary statistics are available.

Definition 2 (Gaussian Copula) *Given a correlation matrix R and a d-dimensional uniform random vector $U = (U_1, \ldots, U_d)$, the Gaussian copula distribution function is defined as:*

$$C_R^{Gauss}(u_1, \ldots, u_d) = \Phi_R\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\right), \tag{2}$$

where Φ_R is the joint cumulative distribution function (CDF) of a multivariate normal distribution with mean zero and covariance matrix R, and Φ^{-1} is the inverse of the standard normal CDF (Haugh, 2016).

Theorem 1 (Sklar's Theorem) *Let H be a joint distribution function with marginals F and G. There exists a copula C such that*

$$H(u, v) = C(F(u), G(v)), \tag{3}$$

for all (u, v) in the support of H (Haugh, 2016; Nelsen, 2006).

Sklar's theorem formally establishes that any multivariate distribution can be expressed in terms of its marginals and a copula, which uniquely determines the dependencies among variables. This result is the basis for copula-driven data reconstruction.

4 Implementation

The Gaussian copula is implemented using the Cholesky decomposition of the correlation matrix R, which yields a lower triangular matrix L such that $R = LL^\top$. This approach is both computationally efficient and numerically stable for generating correlated samples.

The procedure for generating samples is as follows (Haugh, 2016):

1. Perform the Cholesky decomposition of the correlation matrix R:

$$R = LL^\top. \tag{4}$$

2. Generate a vector Z of independent standard normal random variables:

$$Z \sim \mathcal{N}(0, I), \tag{5}$$

where I is the identity matrix.

3. Transform Z using the Cholesky factor L to induce the desired correlation structure:

$$Y = L^\top Z. \tag{6}$$

The resulting vector Y has the desired correlation structure.

4. Apply the standard normal CDF Φ to each component of Y:

$$U_i = \Phi(Y_i), \quad i = 1, \ldots, d. \tag{7}$$

The vector $U = (U_1, \ldots, U_d)$ represents samples from the Gaussian copula.

5. Finally, transform back to the original distributions using the inverse of the marginal CDFs Φ_i^{-1}:

$$X_i = \Phi_i^{-1}(U_i), \quad i = 1, \ldots, d. \tag{8}$$

The vector $X = (X_1, \ldots, X_d)$ represents samples from the original distributions with the desired correlation structure.

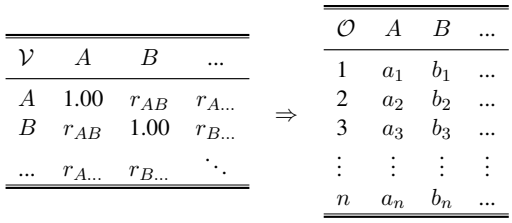

Figure 4: From correlation matrix of variables $\mathcal{V}$ to correlated observations $\mathcal{O}$.

This process ensures that the generated synthetic data both preserves the specified marginal distributions and adheres to the defined correlation structure. The transformation from a correlation matrix to correlated samples is illustrated in Figure 4.

For categorical variables, when means per category are known for another variable (e.g., from ANOVA results), one-hot encoding splits the categorical variable into binary variables. The correlation between mutually exclusive binary categories is determined by their probabilities using $r_{B_iB_j} = -\sqrt{\frac{p_i \cdot p_j}{(1-p_i)(1-p_j)}}$ for $i \neq j$, where p_i and p_j are the marginal probabilities of categories i and j. This ensures realistic negative correlations between exclusive categories, as illustrated in Figure 8. Additionally, positive definiteness of the correlation matrix is ensured by adding a regularization term to the diagonal elements based on the minimum eigenvalue. After reconstruction, the binary indicators are recombined into the original categorical variable, preserving both the categorical structure and correlation patterns.

This approach distributes the original correlation coefficient r_{AB} among the encoded variables proportionally to their category means, enabling integration of grouped statistics such as ANOVA results while maintaining the mutual exclusivity constraint through the probability-based correlation formula.

4.1 Markov Chain Monte Carlo

Definition 3 (Markov Chain Monte Carlo) *Markov Chain Monte Carlo (MCMC) methods are algorithms that sample from a probability distribution by constructing a Markov chain whose equilibrium distribution is the target distribution. Random sampling explores the state space to approximate this distribution (Kauermann et al., 2021).*

Definition 4 (Simulated Annealing) *Simulated annealing is a variant of the Metropolis-Hastings algorithm that introduces a temperature parameter T to control the acceptance probability of candidate states. Provided a suitable cooling schedule, as $T \to 0$ and the number of iterations $t \to \infty$, it converges to the global optimum (Kauermann et al., 2021; Russell and Norvig, 2022).*

Our framework integrates hard constraints within the MCMC procedure to ensure synthetic data samples satisfy specified category counts and value ranges. The state space X consists of summary statistics computed from proposed synthetic datasets. The proposal distribution $q(x'|x)$ generates candidate states in the neighborhood of the current state, incorporating directional shifts based on target statistics such as adjusting means and counts toward desired values. The objective function $f(x)$ quantifies the dissimilarity between current summary statistics and target values, incorporating results from statistical tests, meaning the same statistical tests that have been extracted from the publication are also conducted on the synthetic data during MCMC.

Algorithm 1 Annealing with Metropolis-Hastings

Input: Initial state x_0, temperature T_0, iterations n
Output: Sequence $\{x_t\}_{t=1}^{n}$

$x \leftarrow x_0, T \leftarrow T_0$
for $t = 1$ **to** n **do**
  Propose candidate $x' \sim q(x'|x)$
  Compute objective difference $\Delta \leftarrow f(x') - f(x)$
  Set acceptance probability $p \leftarrow \min(1, e^{-\Delta/T})$
  Draw $u \sim U(0, 1)$
  if $u < p$ **then**
    $x \leftarrow x'$
  end if
  Update temperature $T \leftarrow T_0/\log(1 + t)$
  Store x as x_t
end for

5 Methodology

5.1 Information Extraction

The information extraction pipeline follows a hierarchical strategy that employs chain-of-thought (CoT) prompting with few-shot examples and advanced table recognition. The process begins with TATR (Smock et al., 2022) to convert PDF documents into markdown, thereby preserving the struc-

tural integrity of tables essential for accurate extraction.

After markdown conversion, GPT-4.1 (`gpt-4.1-2025-04-14`) is prompted to identify core dataset characteristics such as sample size (n), variable names, and data types (continuous, ordinal, categorical, binary). This stage establishes the dataset's context and foundational schema. Detailed statistical information is then extracted, including measures of central tendency (mean, median, mode), dispersion (standard deviation, variance, range, interquartile range (IQR)), and distribution (percentiles, confidence intervals). For distributions that deviate from normality, metrics such as skewness and kurtosis are included. Subsequently, the pipeline extracts variable relationships by identifying correlation coefficients and statistical test outcomes (e.g., p-values, effect sizes). In Appendix C,this is explained in more detail. All extracted information is encoded as JSON and validated with pyDantic (Colvin et al., 2025) to ensure compliance with the further processing pipeline.

This staged workflow builds systematically from structural document parsing to detailed statistical and relational characterization, producing robust and reliable data for downstream processing. The use of TATR – with F1 scores of 0.89 - 0.91 in table recognition versus < 0.34 for traditional OCR (Adhikari and Agarwal, 2024) – enhances extraction accuracy. The implementation is extended by the `gmft` package (Wei, 2025), which optimizes native PDF parsing for scientific literature.

5.2 Data Reconstruction

The data reconstruction process integrates analytical and numerical methods to achieve both efficiency and fidelity when generating synthetic datasets. Marginal distributions for each variable are first constructed from the extracted summaries, accounting for both normal and non-normal distributions and respecting constraints such as prescribed ranges or counts.

To capture inter-variable dependencies, extracted correlation coefficients populate a correlation matrix. If explicit coefficients are unavailable, they are derived from reported statistical tests, providing a more complete basis for joint distribution modeling (Harrer et al., 2021; Rosenberg, 2010; Gosling et al., 2024).

For datasets including group-specific statistical tests, categorical variables are temporarily one-hot encoded to binary variables, allowing accurate modeling of grouped marginals. The Gaussian copula is used to generate samples matching the desired correlation structure, and the data are mapped back to their original categorical format after sampling.

The synthetic dataset is further refined using a constrained MCMC approach, ensuring all requirements – such as category frequencies and value ranges – are met as defined by the extracted statistics and test results. This hybrid workflow leverages the scalability of analytical techniques while reserving numerical optimization for the final refinement stage. Multiple candidate datasets are generated analytically, and the minimal objective difference defines the starting point for MCMC.

Overall, by assembling marginals independently and combining them via copula modeling – reserving MCMC for ensuring global constraint satisfaction – this methodology supports scalable, high-fidelity reconstruction of research datasets suitable for large-scale, data-driven applications.

It is not assumed that papers contain all necessary information; instead, the best possible synthetic dataset from the available evidence is reconstructed. This is conducted hierarchical: if a relationship is not described or tested, only independent marginals are generated; if a marginal itself is missing (e.g., counts for categorical data or means for continuous data), that variable is omitted. Defensive try blocks ensure that incomplete extractions do not fail the pipeline, and statistical tests referencing undefined variables are skipped.

5.3 Evaluation

The evaluation of our information extraction and reconstruction pipeline is conducted in two stages to ensure both methodological rigor and practical relevance. In the first stage, we solely assess reconstruction fidelity using manually created, ideal summaries derived from established benchmark datasets (`iris` (Fisher, 1936), `titanic` (Frank E. and Cason, 2017), `tips` (Waskom, 2021), `diamonds` (Wickham, 2016)). This controlled setting enables precise measurement of the pipeline's ability to recover key statistical and structural properties when the ground truth is fully known.

In the second stage, we validate the pipeline on real scientific publications, focusing on seven complex RCTs that provide access to raw tabular data. Prior to evaluation, variables not discussed in the publication or solely used for internal aggrega-

tion (e.g., patient ID, entry date) are removed and variable names are harmonized to match published summaries. Variables that are mentioned in either text, charts or tables, but not further described and explained still remain in the dataset. This end-to-end evaluation examines the framework's robustness and accuracy in realistic, heterogeneous research scenarios, capturing the variability present in actual scientific reporting.

As an additional baseline, we attempted to utilize LLMs to directly generate tabular data from the publications. However, this approach consistently failed to produce valid datasets that comprehensively covered all mentioned variables, adhered to the specified statistical properties, or matched the required dataset length, thereby highlighting the necessity of our structured reconstruction methodology.

5.3.1 Statistical Similarity Metrics

To comprehensively assess reconstruction quality, we quantify it using the following metrics computed between original (X) and reconstructed ($\hat{X}$) data (Kauermann et al., 2021; Mohri et al., 2018).

For basic descriptive alignment, we evaluate the proportion of successfully retrieved variables (variable coverage (**Var. Cov.**)), as well as deviations (in percent) in central tendency (**CT Dev.**) such as mean/median and dispersion measures (**Disp. Dev.**) such as standard deviation/IQR for continuous and ordinal variables, and deviations in category proportions for categorical features (**Cat. Dev.**). These metrics directly measure preservation of key summary statistics and dataset structure.

To assess feature-wise distributional similarity for numerical data, we employ the Kolmogorov-Smirnov (KS) statistic (Massey, 1951):

$$D_{n,m} = \sup_{x \in \mathbb{R}} \left| F_{X,n}(x) - F_{\hat{X},m}(x) \right| \tag{9}$$

where $F_{X,n}$ and $F_{\hat{X},m}$ are empirical CDFs of the original and reconstructed datasets with sample sizes n and m, respectively. The **KS Stat.** measures the maximum difference between these CDFs and is sensitive to differences in distribution shape and location, making it well-suited for detecting discrepancies in continuous variables.

For categorical variables, we use the Jensen-Shannon (JS) divergence based on the Kullback-Leibler (KL) divergence (Lin, 1991):

$$D_{\mathrm{JS}}(p \parallel q) = \frac{1}{2} \left[D_{\mathrm{KL}}(p \parallel m) + D_{\mathrm{KL}}(q \parallel m) \right], \tag{10}$$

where $p_i = p(X_i)$, $q_i = p(\hat{X}_i)$, and $m = \frac{1}{2}(p+q)$, with P_X and $P_{\hat{X}}$ denoting the empirical distributions of a categorical variable in X and $\hat{X}$, respectively. The **JS Div.** ranges from 0 (identical distributions) to 1 (maximal divergence) and is always finite and interpretable, making it appropriate for detecting shifts in class proportions.

To provide a global measure of feature-wise similarity, we report the inverse Kullback-Leibler (InvKL) divergence, which evaluates information loss when approximating the original distribution with the reconstructed one (Kullback and Leibler, 1951). Following (Nguyen et al., 2024), the **InvKL Div.** score is computed as the average over all features:

$$\mathrm{Inv}D_{\mathrm{KL}} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{1 + D_{\mathrm{KL}}(p_i \parallel q_i)} \tag{11}$$

where $p_i = p(X_i)$ and $q_i = p(\hat{X}_i)$ are the distributions of feature i in the original and reconstructed datasets, respectively. This score ranges from 0 (completely different) to 1 (identical), with higher values indicating better similarity. The metric is robust and interpretable, as the KL divergence is finite when the reference distribution is strictly positive, summarizing overall similarity across numerical and categorical features.

Dependency preservation is evaluated by analyzing the deviation of correlation coefficients (**Corr. Dev.**) and statistical test results (including p-values and effect sizes) between the original and reconstructed datasets (**Stat. Dev.**). This ensures maintenance of both linear and nonlinear relationships, which is critical for downstream analyses.

Local data fidelity is further examined via differences in outlier counts (**Outlier Diff.**) and mean record distance (**MR Dist.**), capturing preservation of rare/extreme values and overall multivariate similarity between original and reconstructed samples.

5.3.2 Machine Learning Utility

To assess the practical usefulness of the reconstructed data, we evaluate two aspects (Mohri et al., 2018; Russell and Norvig, 2021).

First, we compute Spearman's ρ correlation between feature importance ranks derived from models trained separately on the original and reconstructed data, assessing preservation of variable importance for prediction (**Feat. Corr.**).

Second, we conduct a Train-on-Real-Test-on-Real (TRTR) vs. Train-on-Synthetic-Test-on-Real (TSTR) comparison and quantify the difference

in predictive performance, specifically F1-score (**F1 Diff.**) and Area Under the Receiver Operating Characteristic Curve (**AUC Diff.**). Models are trained on 70% of each dataset (original and reconstructed) and evaluated on the held-out 30% of the original data. The target variable corresponds to the primary outcome in each publication, typically reflecting group assignment or interventions.

This comprehensive evaluation, grounded in established statistical and machine learning metrics, ensures rigorous validation of both fidelity and utility of the reconstructed data for scientific and practical applications.

6 Results

The reconstruction framework demonstrated robust performance in both controlled experiments and evaluations using real-world publication data. For each dataset, 10 candidate reconstructions were generated, with the best in terms of the objective function refined via $1,500$ iterations of constrained MCMC [2]. The experiments were performed on a MacBook Pro with 24 GB RAM and an Apple M4 Pro chip. All software dependencies are specified via `Poetry` (Eustace, 2018) in the code appendix to ensure reproducibility.

The separate information extraction evaluation in Table 4 and 3 reveals high precision in extracting variable names and description, but lower performance in correctly identifying 'Group Sizes' (F1: [0.75]). A qualitative analysis suggests this is due to the dynamic nature of sample sizes reported in clinical trials (e.g., CONSORT flow diagrams (Falci and Marques, 2015)). For instance, in study DOI ...920511, the algorithm struggled to distinguish between the initial 835 randomized patients and the final 823 analyzed in Group 1. Distinguishing the 'Analyzed' count – which is critical for statistical validation – from the 'Intention-to-Treat' or 'Enrolled' counts remains a key challenge for future improvement.

Figure 5 demonstrates that reconstructions using only mean and standard deviation diverge notably from true distributions for non-normal data (diamonds dataset), while incorporating IQR achieves closer alignment, highlighting the importance of appropriate summary statistics for non-parametric cases. Figure 6 shows close alignment between original and reconstructed petal length distributions, demonstrating preservation of key statistical properties. Figure 7 compares correlation matrices (lower triangle: reconstructed, upper triangle: original) for the iris dataset, indicating strong preservation of joint dependencies.

	iris	titanic	tips	diamonds
Statistical Similarity				
↑ Var. Cov.	5/5	4/4	7/7	7/7
↓ CT Dev.	0.019	0.032	0.021	0.012
↓ Disp. Dev.	0.055	0.030	0.074	0.061
↓ Cat. Dev.	0.000	0.000	0.000	0.000
↓ KS Stat.	0.130	0.170	0.152	0.126
↓ JS Div.	0.000	0.000	0.000	0.000
↑ InvKL Div.	0.947	0.804	0.678	0.889
↓ Corr. Dev.	0.037	0.030	0.104	0.063
↓ Stat. Dev.	0.134	0.346	0.079	0.237
Machine Learning Utility				
↑ Feat. Corr.	1.000	0.800	0.982	0.835
↓ F1 Diff.	0.071	0.098	0.001	0.023
↓ AUC Diff.	0.011	0.090	-0.005	0.002

Table 1: Data Reconstruction Evaluation Metrics Across for a single run on Benchmark Datasets

Quantitative results across synthetic and real publication data for all metrics are shown in Tables 1 and 2. On benchmark datasets with ideal summaries, variable coverage and statistical fidelity are nearly perfect, and the performance gap in downstream ML utility metrics is minimal.

In contrast, real-world datasets reconstructed from scientific publications show notably lower variable coverage and increased deviations in statistical metrics. This reduction is largely due to several factors: variables are omitted entirely from the shared datasets, often for privacy or ethical reasons; variables are insufficiently described or even never mentioned or appear only as aggregated derivatives within the publication; and in some cases, the original raw data provided do not fully correspond to the reported study variables, making reproduction of published results impossible. This performance drop reflects the inherently complex nature of real-world datasets with high dimensionality and intricate relationships poorly documented in publications. Additionally, the absence of statistical tests over complete variable ranges – rather than restricted subsets – limits automated validation. Furthermore, publications frequently report only effect sizes or partial statistical summaries rather than comprehensive test results, which constrains automated reconstruction methods.

Our ablation study (Table 2) demonstrates min-

[2] Hyperparameters were set based on preliminary experiments; as this is a feasibility study, performance optimization was not pursued.

imal performance gaps between our baseline method Text2Tabular and manually annotated Gold Summaries. The contributions of copula modeling and MCMC sampling show reduced impact compared to synthetic datasets due to limited statistical test availability and incomplete documentation. Nevertheless, key downstream metrics – including AUC Difference and Feature Importance Correlation – remain robust, demonstrating the framework's resilience under imperfect and privacy-constrained conditions.

6.1 Limitations

Evaluation was performed only on datasets where both publication and raw data were accessible. In practice, most publications either do not share underlying data, require explicit author contact, or provide unstructured material lacking clear mappings between variables, codebooks, and results. This impedes direct, large-scale evaluation.

Our approach targets well-structured empirical studies with sufficient reporting for extraction and synthesis, ideally following standards (e.g., CONSORT, STROBE) (Grech and Eldawlatly, 2024). The current evaluation is limited to seven RCTs, covering diverse publishers, layouts, writing styles, domains, and experimental designs, and we are extending it to non-RCT studies where suitable open-access raw data are available.

The comparison of extracted versus hand-curated JSON summaries confirms high accuracy for described variables, though occasional splitting of categorical variables into binary ones or misclassification of ordinal as continuous and vice versa was observed. Aggregations pose a further challenge: unless variables are explicitly summarized and described (e.g., for ANOVA), reconstruction of some features is impossible. Thus, for poorly described publications, Text2Tabular does not recover all variables automatically.

In terms of risks, while the framework is designed to generate datasets that reflect published summaries, there remains a risk of inadvertently generating synthetic data that could be misinterpreted as real patient data. To mitigate this, we emphasize that all generated datasets are synthetic and should not be used for clinical decision-making. Further, synthetic data generated by Text2Tabular will inherently preserve biases present in the original data and reported statistics.

7 Conclusion

Text2Tabular introduces the first unified NLP-to-synthesis pipeline capable of automatically reconstructing tabular datasets directly from natural language descriptions in scientific publications. This approach represents a fundamentally novel contribution, as no existing work has attempted to bridge textual statistical reporting and automated data synthesis at this scale. The framework effectively preserves essential distributional properties and relationships across diverse variable types. Our results demonstrate the tool's ability to generate statistically valid datasets that reproduce outcomes of common statistical tests, enabling further analyses and meta-studies when raw data is inaccessible. By combining analytical reconstruction with constrained MCMC refinement, Text2Tabular balances efficiency and accuracy for large-scale data synthesis. This approach fosters reproducibility and transparency while facilitating secondary analyses in privacy-constrained domains. Future work can leverage Text2Tabular as a foundation to train more capable models that capture nuanced subtleties in data descriptions and further bridge the gap between published evidence and accessible data.

Acknowledgements

This research has been supported by the German Federal Ministry of Education and Research (BMBF) grant 16IS23069 Software Campus 3.0 (TU München). We would like to thank the anonymous reviewers for their valuable feedback.

References

2022. How to choose an appropriate statistical test.

Narayan S. Adhikari and Shradha Agarwal. 2024. A Comparative Study of PDF Parsing Tools Across Diverse Document Categories. *Preprint*, arXiv:2410.09871.

Peter Armitage, Geoffrey Berry, and J. N. S. Matthews. 2013. *Statistical Methods in Medical Research*. John Wiley & Sons.

[1] (Solnick et al., 2020; Peyton, 2020)
[2] (Etyang et al., 2019, 2023)
[3] (Magai et al., 2019b,a)
[4] (Rosen et al., 2016; Maskew et al., 2020)
[5] (Xu et al., 2019; Xu, 2018)
[6] (Jordans et al., 2021b,a)
[7] (Hull et al., 2024; Reynolds, 2023)
[8] Incomplete data, not all labels of target variables available due to data custodianship fragmentation.

Metric	Approach	Publication DOI (truncated)						
		...920511[1]	...011771[2]	...946322[3]	...002015[4]	...002785[5]	...003621[6]	...000028[7]
Statistical Similarity								
↑ **Var. Cov.**	Baseline (T2T)	9/66	15/25	10/14	4/18	24/30	33/50	9/12
	Gold Summaries	10/66	17/25	10/14	4/18	25/30	33/50	9/12
↓ **CT Dev.**	Marginals Only	0.113	0.087	0.140	0.143	0.814	0.124	0.254
	Copula Only	0.113	0.086	0.142	0.151	0.814	0.124	0.258
	Baseline (T2T)	0.110	0.086	0.093	0.252	0.816	0.127	0.252
	Gold Summaries	0.109	0.123	0.121	0.145	0.829	0.117	0.255
↓ **Disp. Dev.**	Baseline (T2T)	0.355	0.417	1.622	1.037	0.955	0.091	0.491
↓ **Cat. Dev.**	Baseline (T2T)	0.463	0.773	0.092	0.259	0.152	0.014	0.566
	Gold Summaries	0.382	0.585	0.005	0.259	0.152	0.014	0.566
↓ **KS Stat.**	Baseline (T2T)	0.131	0.144	0.111	0.377	0.229	0.051	0.309
↓ **JS Div.**	Baseline (T2T)	1.000	0.568	0.030	0.000	0.002	0.001	0.722
↑ **InvKL Div.**	Marginals Only	0.612	0.876	0.688	0.644	0.357	0.828	0.206
	Copula Only	0.612	0.876	0.683	0.633	0.357	0.828	0.202
	Baseline (T2T)	0.564	0.876	0.892	0.682	0.357	0.780	0.234
	Gold Summaries	0.566	0.855	0.735	0.674	0.382	0.780	0.110
↓ **Corr. Dev.**	Marginals Only	0.118	0.162	0.160	0.115	0.115	0.094	0.388
	Copula Only	0.118	0.162	0.169	0.107	0.113	0.094	0.479
	Baseline (T2T)	0.120	0.162	0.179	0.105	0.110	0.095	0.348
	Gold Summaries	0.099	0.160	0.143	0.102	0.115	0.095	0.373
↓ **Stat. Dev.**	Marginals Only			47.292		0.473		
	Copula Only			46.670		0.476		
	Baseline (T2T)			44.334		0.460		
	Gold Summaries			34.370		0.352		
Machine Learning Utility								
↑ **Feat. Corr.**	Marginals Only	0.714	[8]	0.667	1.000	0.804	0.740	0.762
	Copula Only	0.762	[8]	0.683	1.000	0.857	0.827	0.214
	Baseline (T2T)	0.571	[8]	0.733	1.000	0.772	0.798	0.405
	Gold Summaries	0.583	[8]	0.433	1.000	0.860	0.793	0.524
↓ **F1 Diff.**	Baseline (T2T)	0.230	[8]	0.033	0.146	0.140	0.261	0.245
↓ **AUC Diff.**	Marginals Only	-0.004	[8]	0.104	0.046	-0.053	0.221	0.021
	Copula Only	0.007	[8]	0.011	0.028	-0.037	0.164	0.058
	Baseline (T2T)	0.000	[8]	-0.036	0.062	0.137	0.299	0.009
	Gold Summaries	0.013	[8]	-0.079	0.012	0.100	0.245	0.028

Table 2: Ablation Study for a single run: Data Reconstruction Evaluation for real-world publications (truncated DOI number as columns). We evaluate each pipeline component: (1) Marginals Only (baseline distributions), (2) + Copula (dependency structure), (3) + MCMC (complete Text2Tabular incl. extracted summaries), and (4) Text2Tabular with Gold Summaries. Abbreviated metrics explained in Section 5.3.

Samuel Colvin, Eric Jolibois, and Hasan Ramezani. 2025. Pydantic/pydantic: Data validation using Python type hints.

Samantha Duchscherer, Robert Stewart, and Marie Urban. 2019. Revengc: An R package to Reverse Engineer Summarized Data. *The R Journal*, 10(2):114.

Anthony O. Etyang, Sailoki Kapesa, Emily Odipo, Evasius Bauni, Catherine Kyobutungi, Marwah Abdalla, Paul Muntner, Solomon K. Musani, Alex Macharia, Thomas N. Williams, J. Kennedy Cruickshank, Liam Smeeth, and J. Anthony G. Scott. 2019. Effect of Previous Exposure to Malaria on Blood Pressure in Kilifi, Kenya: A Mendelian Randomization Study. *Journal of the American Heart Association: Cardiovascular and Cerebrovascular Disease*, 8(6):e011771.

Anthony O. Etyang, Sailoki Kapesa, Emily Odipo, Evasius Bauni, Catherine Kyobutungi, Marwah Abdalla, Paul Mutner, Solomon K. Musani, Alex Macharia, Thomas N. Williams, J Kennedy Cruickshank, Liam Smeeth, and J Anthony G. Scott. 2023. Data from: Effect of previous exposure to malaria on blood pressure in kilifi, kenya: A mendelian randomization study.

Sébastien Eustace. 2018. Poetry: Python packaging and dependency management made easy.

Saulo Gabriel Moreira Falci and Leandro Silva Marques. 2015. CONSORT: When and how to use it. *Dental Press Journal of Orthodontics*, 20(3):13–15.

Guoshuang Feng, Guoyou Qin, Tao Zhang, Zheng Chen, and Yang Zhao. 2022. Common Statistical Methods and Reporting of Results in Medical Research. *Cardiovascular Innovations and Applications*, 6(3).

R. A. Fisher. 1936. Iris. UCI Machine Learning Repository.

Harrell Jr. Frank E. and Thomas Cason. 2017. Titanic dataset.

Corentin J Gosling, Samuele Cortese, Marco Solmi, Belen Haza, Eduard Vieta, Richard Delorme, Paolo Fusar-Poli, and Joaquim Radua. 2024. *An Automatic Suite for Estimation of 11 Different Effect Size Measures and Flexible Conversion across Them*.

Victor Grech and Abdelazeem A. Eldawlatly. 2024. STROBE, CONSORT, PRISMA, MOOSE, STARD, SPIRIT, and other guidelines – Overview and application. *Saudi Journal of Anaesthesia*, 18(1):137–141.

Mathias Harrer, Pim Cuijpers, Toshi Furukawa, and David Ebert. 2021. *Doing Meta-Analysis with R: A Hands-On Guide*. Chapman and Hall/CRC, Boca Raton London New York.

Martin Haugh. 2016. An Introduction to Copulas.

James A. Heathers, Jordan Anaya, Tim van der Zee, and Nicholas JL Brown. 2018. Recovering data from summary statistics: Sample Parameter Reconstruction via Iterative TEchniques (SPRITE). Technical Report e26968v1, PeerJ Preprints.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations 2023*.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. 2025. Accurate predictions on small data with a tabular foundation model. *Nature*.

Margaret A Hull, Elizabeth A Nunamaker, and Penny S Reynolds. 2024. Effects of Refined Handling on Reproductive Indices of BALB/cJ and CD-1 IGS Mice. *Journal of the American Association for Laboratory Animal Science : JAALAS*, 63(1):3–9.

Nicklas Jävergård, Rainey Lyons, Adrian Muntean, and Jonas Forsman. 2024. Preserving correlations: A statistical method for generating synthetic data. *Preprint*, arXiv:2403.01471.

Mark J. D. Jordans, Brandon A. Kohrt, Manaswi Sangraula, Elizabeth L. Turner, Xueqi Wang, Pragya Shrestha, Renasha Ghimire, Edith van't Hof, Richard A. Bryant, Katie S. Dawson, Kedar Marahatta, Nagendra P. Luitel, and Mark van Ommeren. 2021a. Data from: Effectiveness of Group Problem Management Plus, a brief psychological intervention for adults affected by humanitarian disasters in Nepal: A cluster randomized controlled trial.

Mark J. D. Jordans, Brandon A. Kohrt, Manaswi Sangraula, Elizabeth L. Turner, Xueqi Wang, Pragya Shrestha, Renasha Ghimire, Edith van't Hof, Richard A. Bryant, Katie S. Dawson, Kedar Marahatta, Nagendra P. Luitel, and Mark van Ommeren. 2021b. Effectiveness of Group Problem Management Plus, a brief psychological intervention for adults affected by humanitarian disasters in Nepal: A cluster randomized controlled trial. *PLOS Medicine*, 18(6):e1003621.

Göran Kauermann, Helmut Küchenhoff, and Christian Heumann. 2021. *Statistical Foundations, Reasoning and Inference: For Science and Data Science*, 1st ed. 2021 edition edition. Springer, Cham.

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37:145–151.

Dorcas N. Magai, Michael Mwaniki, Amina Abubakar, Shebe Mohammed, Anne L. Gordon, Raphael Kalu, Paul Mwangi, Hans M. Koot, and Charles R. Newton. 2019a. Data from: A Randomized Control Trial of Phototherapy and 20\% Albumin Versus Phototherapy and Saline in Kilifi, Kenya.

Dorcas N. Magai, Michael Mwaniki, Amina Abubakar, Shebe Mohammed, Anne L. Gordon, Raphael Kalu, Paul Mwangi, Hans M. Koot, and Charles R. Newton. 2019b. A randomized control trial of phototherapy and 20% albumin versus phototherapy and saline in Kilifi, Kenya. *BMC Research Notes*, 12:617.

Mhairi Maskew, Sydney Rose, and Matthew Fox. 2020. Data from: Initiating Antiretroviral Therapy for HIV at a Patient's First Clinic Visit: The RapIT Randomized Controlled Trial.

Frank J. Massey. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78.

Prabhaker Mishra, Chandra Mani Pandey, Uttam Singh, Amit Keshri, and Mayilvaganan Sabaretnam. 2019. Selection of Appropriate Statistical Methods for Data Analysis. *Annals of Cardiac Anaesthesia*, 22(3):297.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning*, 2 edition. Adaptive Computation and Machine Learning Series. MIT Press, London, England.

Shinpei Nakamura-Sakai, Fadi Hamad, Saheed Obitayo, and Vamsi K. Potluru. 2024. A supervised generative optimization approach for tabular data. *Preprint*, arXiv:2309.05079.

Roger B. Nelsen. 2006. *An Introduction to Copulas*, second edition. Springer Series in Statistics. Springer, New York, NY.

Dang Nguyen, Sunil Gupta, Kien Do, Thin Nguyen, and Svetha Venkatesh. 2024. Generating Realistic Tabular Data with Large Language Models. In *2024 IEEE*

International Conference on Data Mining (ICDM), pages 330–339. IEEE Computer Society.

Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The Synthetic Data Vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, Montreal, QC, Canada. IEEE.

Kyle Peyton. 2020. Data from: Effect of physician gender and race on simulated patients' ratings and confidence in their physicians: A randomized clinical trial.

Maciej P. Polak and Dane Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, 15(1):1569.

Richeek Pradhan, David C. Hoaglin, Matthew Cornell, Weisong Liu, Victoria Wang, and Hong Yu. 2019. Automatic extraction of quantitative data from ClinicalTrials.gov to conduct meta-analyses. *Journal of clinical epidemiology*, 105:92–100.

Penelope Reynolds. 2023. Data from: Effects of refined handling on reproductive indices of BALB/cJ and CD-1 IGS mice.

Sydney Rosen, Mhairi Maskew, Matthew P. Fox, Cynthia Nyoni, Constance Mongwenyana, Given Malete, Ian Sanne, Dorah Bokaba, Celeste Sauls, Julia Rohr, and Lawrence Long. 2016. Initiating Antiretroviral Therapy for HIV at a Patient's First Clinic Visit: The RapIT Randomized Controlled Trial. *PLOS Medicine*, 13(5):e1002015.

Michael S. Rosenberg. 2010. A Generalized Formula for Converting Chi-Square Tests to Effect Sizes for Meta-Analysis. *PLOS ONE*, 5(4):e10059.

Stuart J. Russell and Peter Norvig. 2022. *Artificial Intelligence: A Modern Approach*, fourth edition, global edition edition. Prentice Hall Series in Artificial Intelligence. Pearson, Boston.

Stuart Jonathan Russell and Peter Norvig. 2021. *Artificial Intelligence – a Modern Approach, Global Edition*, 4 edition. Pearson Education. Prentice Hall, Harlow, United Kingdom.

Ruxue Shi, Yili Wang, Mengnan Du, Xu Shen, and Xin Wang. 2025. A Comprehensive Survey of Synthetic Tabular Data Generation. *Preprint*, arXiv:2504.16506.

Brandon Smock, Rohith Pesala, and Robin Abraham. 2022. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4624–4632, New Orleans, LA, USA. IEEE.

Rachel E. Solnick, Kyle Peyton, Gordon Kraft-Todd, and Basmah Safdar. 2020. Effect of Physician Gender and Race on Simulated Patients' Ratings and Confidence in Their Physicians: A Randomized Trial. *JAMA Network Open*, 3(2):e1920511.

Michael L. Waskom. 2021. Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Galen Wei. 2025. Gmft.

Hadley Wickham. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Dong Xu. 2018. Data from: Lay health supporters aided by mobile text messaging to improve adherence, symptoms, and functioning among people with schizophrenia in a resource-poor community in rural China (LEAN): A randomized controlled trial.

Dong (Roman) Xu, Shuiyuan Xiao, Hua He, Eric D. Caine, Stephen Gloyd, Jane Simoni, James P. Hughes, Juan Nie, Meijuan Lin, Wenjun He, Yeqing Yuan, and Wenjie Gong. 2019. Lay health supporters aided by mobile text messaging to improve adherence, symptoms, and functioning among people with schizophrenia in a resource-poor community in rural China (LEAN): A randomized controlled trial. *PLOS Medicine*, 16(4):e1002785.

Fengxia Yan, Mayberry Robert, and Yonggang Li. 2017. Statistical methods and common problems in medical or biomedical science research. *International Journal of Physiology, Pathophysiology and Pharmacology*, 9(5):157–163.

A Additional Visualizations

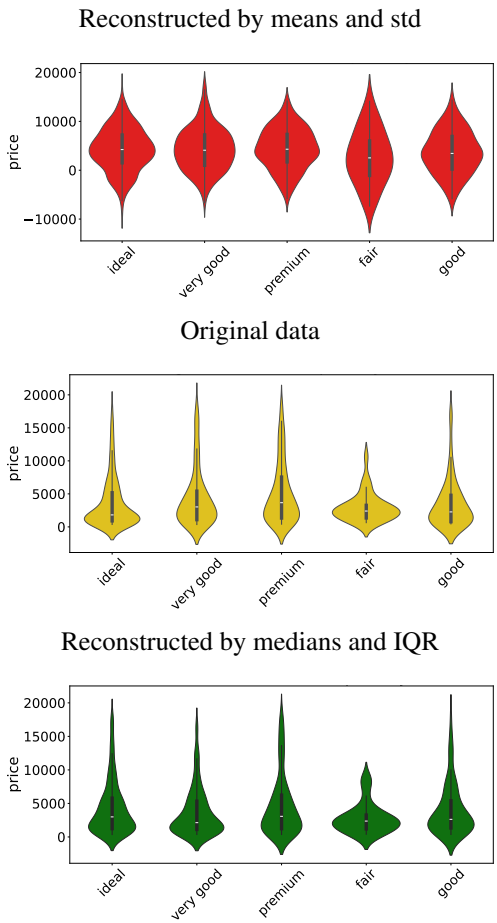

Figure 5: Effects of including IQR for non-normal data in an ANOVA example. Top: reconstruction by means/std; middle: original; bottom: reconstruction by medians/IQR (diamonds dataset, price by cut).

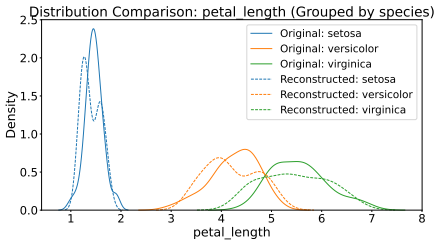

Figure 6: Close distributions of reconstructed and original petal length in the iris dataset.

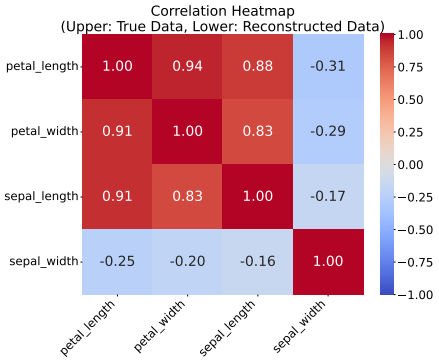

Figure 7: Heatmaps comparing correlation structures of reconstructed (lower triangle) and original (upper triangle) iris data.

$$
\dots \Rightarrow
\begin{array}{c|cccc}
\mathcal{V} & A & B_1 & B_2 & \dots \\
\hline
A & 1.0 & r_{AB_1} & r_{AB_2} & r_{A\dots} \\
B_1 & r_{AB_1} & 1.0 & -r_{B_1B_2} & r_{B\dots} \\
B_2 & r_{AB_2} & -r_{B_2B_1} & 1.0 & r_{B\dots} \\
B_3 & r_{AB_3} & -r_{B_3B_1} & -r_{B_3B_2} & r_{B\dots} \\
\dots & r_{A\dots} & r_{B\dots} & r_{B\dots} & \ddots
\end{array}
\Rightarrow \dots
$$

$$
\dots \Rightarrow
\begin{array}{c|cccc}
\mathcal{O} & A & B_1 & B_2 & \dots \\
\hline
1 & a_1 & b_{11} & b_{21} & \dots \\
2 & a_2 & b_{12} & b_{22} & \dots \\
3 & a_3 & b_{13} & b_{23} & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
n & a_n & b_{1n} & b_{2n} & \dots
\end{array}
\Rightarrow \dots
$$

Figure 8: Splitting a categorical variable into one-hot encoded variables and distributing correlation coefficients.

B Information Extraction

DOI	Category	F1	Prec	Rec
...920511	**Total**	0.762	0.762	0.762
	Sample Size	1.000	1.000	1.000
	Group Names	1.000	1.000	1.000
	Group Sizes	0.250	0.250	0.250
	Variable Names	0.800	0.769	0.833
	Variable Content	0.767	0.767	0.767
	Statistical Tests	1.000	1.000	1.000
...011771	**Total**	0.647	0.628	0.667
	Sample Size	1.000	1.000	1.000
	Group Names	1.000	1.000	1.000
	Group Sizes	1.000	1.000	1.000
	Variable Names	0.875	0.933	0.824
	Variable Content	0.673	0.651	0.696
	Statistical Tests	0.000	0.000	0.000
...946322	**Total**	0.980	0.980	0.980
	Sample Size	0.000	0.000	0.000
	Group Names	1.000	1.000	1.000
	Group Sizes	0.000	0.000	0.000
	Variable Names	1.000	1.000	1.000
	Variable Content	1.000	1.000	1.000
	Statistical Tests	1.000	1.000	1.000
...002015	**Total**	0.920	0.896	0.945
	Sample Size	1.000	1.000	1.000
	Group Names	1.000	1.000	1.000
	Group Sizes	1.000	1.000	1.000
	Variable Names	0.906	0.857	0.960
	Variable Content	0.905	0.877	0.934
	Statistical Tests	1.000	1.000	1.000
...002785	**Total**	0.928	0.981	0.881
	Sample Size	1.000	1.000	1.000
	Group Names	1.000	1.000	1.000
	Group Sizes	1.000	1.000	1.000
	Variable Names	0.917	1.000	0.846
	Variable Content	0.900	0.973	0.837
	Statistical Tests	1.000	1.000	1.000
...003621	**Total**	0.950	0.938	0.962
	Sample Size	1.000	1.000	1.000
	Group Names	1.000	1.000	1.000
	Group Sizes	1.000	1.000	1.000
	Variable Names	0.936	0.898	0.978
	Variable Content	0.939	0.925	0.954
	Statistical Tests	1.000	1.000	1.000
...000028	**Total**	0.959	0.921	1.000
	Sample Size	1.000	1.000	1.000
	Group Names	1.000	1.000	1.000
	Group Sizes	1.000	1.000	1.000
	Variable Names	0.957	0.917	1.000
	Variable Content	0.933	0.875	1.000
	Statistical Tests	1.000	1.000	1.000

Table 3: Detailed extraction performance per DOI.

Category	Avg F1	Avg Prec	Avg Rec
Total	0.878	0.872	0.885
Sample Size	0.857	0.857	0.857
Group Names	1.000	1.000	1.000
Group Sizes	0.750	0.750	0.750
Variable Names	0.913	0.911	0.920
Variable Content	0.874	0.867	0.884
Statistical Tests	0.857	0.857	0.857

Table 4: Average extraction performance across all DOIs.

C Prompts Used for Automated Data Extraction

This appendix documents the exact prompts and methodology used in the Text2Tabular extraction pipeline. The extraction process leverages **chain-of-thought prompting**, **multi-shot (few-shot) examples**, and **temperature=0** for deterministic outputs. All extracted data is validated using `pydantic` models to ensure structural and semantic correctness.

The pipeline is hierarchical: each step builds on the previous, with outputs from one prompt feeding into the next. Prompts are designed to be explicit, robust, and reproducible, and include both instructions and concrete input/output examples.

C.1 Pipeline Overview

- **Chain-of-thought prompting:** Each prompt guides the LLM through a sequence of reasoning steps, ensuring context is preserved and extraction is systematic.
- **Multi-shot (few-shot) learning:** Each prompt includes multiple input/output examples to anchor the LLM's behavior and improve reliability.
- **Temperature:** All LLM calls use `temperature=0.0` to maximize determinism and reproducibility.
- **Validation:** All outputs are parsed and validated using `pydantic` models, enforcing strict adherence to the expected schema.

C.2 Study Information Extraction

Instruction: *You are a specialized statistical extractor focusing ONLY on identifying core study parameters. Focus on tables, since they dictate the structure of the data. The groups are often defined as column headers in tables or explicitly mentioned in the text.*

Extract the following information from the research paper:

1. *study_size (total number of participants)*
2. *groups (e.g., "Drug intervention group", "Control group", etc.)*
3. *group_sizes (number of participants in each group)*

Return the data in the following JSON format:

```
{
    "study_size": int,
    "groups": [string, string, ...],
    "group_sizes": {"group_name": int,
     ...}
}
```

When extracting group information, always use the most granular (detailed) subgroup definitions available from the text or tables. If a value (e.g., group_size) is only reported for a broader (meta) group, assign that value to all its subgroups. Ensure that the sum of all group_sizes equals the study_size. If any information is missing, use None. Aim for short and concise group names. If the group name is too long, use an abbreviation or a short form.

—

Preview of the next step: In the next step, you will be asked to extract all variables measured in the study and categorize them by type (continuous, ordinal, categorical, binary), using the group names and sizes you provide here.

Example:

```
Input:
| Characteristic | Treatment A (n=45)
    | Treatment B (n=38) | Control (n
    =42) |
|----------------|-------------------|
     Age, mean (SD) | 64.2 (12.1)
     | 62.8 (11.9)      | 63.1 (12.4)
      |
| BMI, kg/m^2     | 28.1 +/- 4.2
      | 27.9 +/- 3.8        | 28.5 +/-
     4.1     |

Output:
{
    "study_size": 125,
    "groups": ["Treatment A", "
    Treatment B", "Control"],
    "group_sizes": {
        "Treatment A": 45,
        "Treatment B": 38,
        "Control": 42
    }
}
```

C.3 Variable Extraction and Categorization

Instruction: *You are a specialized statistical extractor focusing ONLY on identifying variables. Based on the previously extracted study parameters, extract all variables and categorize them by type. Focus on tables to identify variables, as they often provide the most structured information.*

Extract ONLY the variable names categorized as:

- ***Continuous variables:*** *measured on a numerical, truly continuous scale (e.g., Weight, Height, BMI)*
- ***Ordinal variables:*** *measured on a numerical, continuous scale but can only take discrete values (e.g., Age, Ratings)*
- ***Binary variables:*** *represent a YES/NO, TRUE/FALSE, or SUCCESS/FAILURE outcome (e.g., Mortality, Smoker)*
- ***Categorical variables:*** *with distinct, named categories that are not binary outcomes (e.g., Race, Education level)*

RULE: In continuous and ordinal variables, distributions are described by mean, std, median, IQR, etc., and there are no counts. Counts only occur in binary and categorical variables. When in doubt between categorical and binary, choose binary. The refinement step will consolidate related binary variables into categorical ones.

Return the data in the following JSON format:

```
{
    "variables": {
        "continuous": [list of
    variable names],
        "ordinal": [list of variable
    names],
        "binary": [list of variable
    names],
        "categorical": [list of
    variable names]
    }
}
```

Example:

```
Input:
| Variable | Treatment (n=50) |
    Control (n=48) |
|----------|----------------|-------|
| Age, years | 65.2 +/- 8.4 | 64.1 +/-
     9.2 |
| BMI, kg/m^2 | 28.5 +/- 4.1 | 27.9 +/-
     3.8 |
| Gender, Female | 28 (56%) | 25 (52%)
     |
| Mortality | 7 (14%) | 12 (25%) |
| Education level |  |  |
| - High school | 18 (36%) | 20 (42%)
    |
| - College | 20 (40%) | 18 (38%) |
| - Graduate | 12 (24%) | 10 (21%) |

Output:
{
    "variables": {
        "continuous": ["BMI"],
        "ordinal": ["Age"],
        "binary": ["Gender (Female)",
    "Mortality"],
        "categorical": ["Education
    level"]
    }
}
```

C.4 Statistical Value Extraction

Instruction: *You are a specialized statistical extractor focusing ONLY on extracting statistical values. For each variable in each group, extract the appropriate statistical information. Keep the provided JSON structure intact and fill in the values based on the information provided in the document. Fill in the values for each variable type for ALL retrieved groups. If you have only values for the total study, use the SAME values for ALL groups. If you have no values for a group, leave it empty.*

Continuous/Ordinal: mean, std, median, IQR, min, max, etc.
Binary: count and denominator per group
Categorical: count per category and group, plus group total

Return in valid JSON format matching the variable structure, with each variable containing the appropriate statistical information for each group. DO NOT use any other structure than the one for its dedicated variable type.

Example:

```
Input:
| Variable | Total Study (N=120) |
|----------|---------------------|
| BMI, kg/m^2 | 28.2 +/- 4.0 |
| Age, years | 65.5 +/- 8.2 |

Output:
{
    "variables": {
        "continuous": {
            "BMI": {
                "Treatment A": {"mean":
     28.2, "std": 4.0},
                "Treatment B": {"mean":
     28.2, "std": 4.0},
                "Control": {"mean":
    28.2, "std": 4.0}
            }
        },
        "ordinal": {
            "Age": {
                "Treatment A": {"mean":
     65.5, "std": 8.2},
                "Treatment B": {"mean":
     65.5, "std": 8.2},
                "Control": {"mean":
    65.5, "std": 8.2}
            }
        }
    }
}
```

C.5 Variable Refinement

Instruction: *You are a specialized statistical extractor. STRICTLY follow these rules:*

1. *If a variable's data does not fit the template for its assigned type, MOVE it to the correct type.*
2. *Do NOT invent or modify field names.*
3. *For each variable, use ONLY the structure shown in the templates below for its type. Use null for missing or unknown values.*
4. *Consolidate mutually exclusive binary variables into a categorical variable if appropriate.*
5. *Validate your output: every variable must match the template for its type.*
6. *Use ONLY the group names previously extracted.*

Return ONLY valid JSON matching these examples, using the ACTUAL and PRECISE variable and group names from the study. Do not include any explanatory text.

Example:

```
Input:
{
    "binary": {
        "Married": {"Group A": {"count
    ": 25, "denominator": 50}},
        "Single": {"Group A": {"count":
     15, "denominator": 50}},
        "Divorced": {"Group A": {"
    count": 10, "denominator": 50}}
    }
}

Output:
{
    "variables": {
        "binary": {},
        "categorical": {
            "Marital Status": {
                "Group A": {
                    "Married": 25,
                    "Single": 15,
                    "Divorced": 10,
                    "total": 50
                }
            }
        }
    }
}
```

C.6 Statistical Test Extraction

Instruction: *You are a specialized statistical extractor focusing ONLY on extracting statistical tests.*

For each statistical test mentioned in the document, extract:

1. *The variables involved (names must match previously identified variables)*
2. *The type of test (must be one of: pearson, spearman, chi_square, mcnemar, wilcoxon_signed_rank, paired_t_test, wilcoxon_mann_whitney, unpaired_t_test, friedman, kruskal_wallis, one_way_anova, ranova)*
3. *The test statistic value*
4. *The p-value*
5. *Any effect size mentioned*
6. *The groups involved (only use groups previously identified in the study)*

If group names in the text do not match exactly, map them to the closest previously extracted group name or use "Unknown Group" if no reasonable match exists. Every test must use only groups from the extracted list.

Return in the following JSON format:

```
{
    "statistical_tests": [
        {
            "variables": ["variable1",
     "variable2"],
            "test_type": "
    valid_test_type_from_list_above",
            "test_statistic":
    float_value,
            "p_value": float_value,
            "effect_size":
    optional_float_value,
            "group_means":
    optional_list_of_means,
            "groups": ["group1", "
    group2"]
        },
        ...
    ]
}
```

Example:

```
Input:
Baseline characteristics were compared
     using independent t-tests for
    continuous
variables and chi-square tests for
    categorical variables. The mean
    age was
significantly different between groups
     (t=2.45, p=0.016). Gender
    distribution
did not differ significantly between
    groups (Chi^2 =1.23, p=0.267).

Output:
{
    "statistical_tests": [
        {
            "variables": ["Age"],
            "test_type": "
    unpaired_t_test",
            "test_statistic": 2.45,
            "p_value": 0.016,
            "effect_size": null,
            "group_means": null,
            "groups": ["Treatment", "
    Control"]
        },
        {
            "variables": ["Gender (
    Female)"],
            "test_type": "chi_square",
            "test_statistic": 1.23,
            "p_value": 0.267,
            "effect_size": null,
            "group_means": null,
            "groups": ["Treatment", "
    Control"]
        }
    ]
}
```

Note: For brevity, only the main structure of each prompt and a single example are shown here. Full prompts, including all instructions and additional examples, are available in the codebase.