Data Driven Design as a Challenge Task for Few- and Zero-Shot Information Extraction

Sireesh Gururaja[1] **Jeremiah Milbauer**[1] **Hung-Yi Lin**[2]
Anthony Rollett[2] **Emma Strubell**[1,2]
[1]Language Technologies Institute, School of Computer Science
[2]Department of Materials Science and Engineering
Carnegie Mellon University
sgururaj@cs.cmu.edu

Abstract

In this work we present a challenge dataset for few- and zero-shot multimodal information extraction to support the data-driven design (DDD) of materials. The benchmark repurposes manually-verified tabular data from Jensen et al. (2019)'s study of zeolite synthesis. The proposed dataset is intended to evaluate systems' capabilities in information extraction, disambiguation, and normalization from tables and related text (e.g. captions), in both multimodal and text-only settings. We argue that data-driven design presents a promising task — data-rich, useful, and challenging — against which to benchmark next-generation information extraction systems.

1 Introduction

Data-driven design (DDD), a process by which materials scientists use information extracted from the literature to inform future experiments, has emerged in the past decade as an important method by which to accelerate the discovery of materials (Olivetti et al., 2020). As NLP methods have evolved, so too has their application to data-driven design problems, from pipeline-based approaches using multiple purpose-trained models and relying heavily on rules-based, handwritten heuristics (Kim et al., 2017; Court and Cole, 2018; Jensen et al., 2019, *inter alia*) to end-to-end approaches involving fine-tuning large language models (LLMs) to act as information extractors and assistants (Zheng et al., 2023), or generate structured output describing properties directly (Dagdelen et al., 2024).

However, even current data-driven design work relies on annotated data. The method proposed in Dagdelen et al. (2024), for instance, suggests annotating " 100–500 text passages" in order to fine-tune an LLM to produce structured data. This type of data can be difficult to produce: it often requires domain expertise to collect, verify, and postprocess into a format that is appropriate for training such models. This problem is exacerbated when considering that data-driven design efforts often seek to extract information into specific, non-overlapping schemas, limiting the possibility of data sharing or transfer learning to reduce the burden on individual materials scientists seeking to start a new data-driven design project.

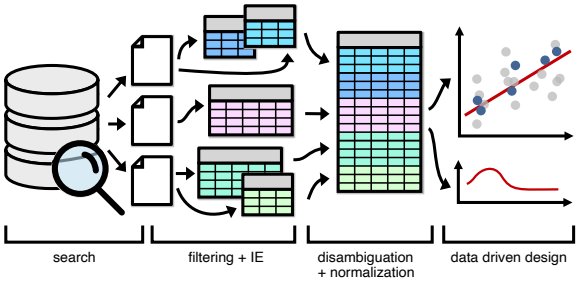

Figure 1: The process of data-driven design. Our benchmark focuses on the middle two phases: extraction/filtering and disambiguation/normalization

Given, however, the rapid development of models that can process scientific documents, both in text-only and multimodal formats, we view the possibility of data-driven design projects that require little to no annotated data as both highly desirable and feasible in the near future. The extraction challenges created by typical data-driven design projects also remain at the frontier of the capabilities of even newer models: processing visually-rich documents with information in text, tables, and figures; disambiguating extracted information to a standardized schema; and performing consistent numerical reasoning to normalize scales and units so that extracted information is comparable across papers (Miret and Krishnan, 2024).

In this paper, we propose an initial dataset to demonstrate DDD's suitability as a challenge task and benchmark for next-generation information extraction models, focused on replicating a subset of the data extracted in Jensen et al. (2019), which fo-

cuses on zeolite synthesis. Zeolites are crystalline materials with a variety of industrial uses, and have therefore been the subject of many studies in DDD. We intend for this benchmark to reflect a realistic subset of the data-driven design process, while at the same time making some allowances for model affordances like context windows and expected input modalities.

We therefore present the benchmark in two settings: a multimodal variant, which presents as input an image or a PDF page containing the information to extract, and a text-only variant, in which input is XML/HTML containing the same information. In both cases, the expected output is the disambiguated, normalized table presented in the original paper, which unifies the data from across many documents. We propose zero-shot baselines in both settings, and find that while modern systems perform strongly on common formats of tables, their ability to extract and integrate information varies widely between different sources of information and different table layouts.

We are currently working to determine to determine in what format we can release this dataset, but note that the multimodal setting of the benchmark can be easily regenerated given the downloaded PDFs and the metadata that we provide. We will release a script to perform this reconstruction.

2 Data Driven Design and Task Scoping

In keeping with the literature, we conceptualize of DDD as a task separated into four phases, which we visualize in figure 1, and discuss below:

- **Search/retrieval** In this phase, researchers typically collect a large number of papers using high-recall, low-precision methods like keyword matching. Papers are typically downloaded in a number of different formats, including scraped HTML, XML from APIs, and PDFs that are then converted to text.
- **Information extraction and filtering** In this phase, researchers will attempt to extract information corresponding to the schema of interest from the retrieved papers. Notably, in this phase, not all extracted information is relevant, necessitating a filtering process. The specific methods by which this phase is carried out have varied over time. Olivetti et al. (2020) describe an pipelined approach common at the time; end-to-end approaches have since become more popular.
- **Disambiguation and normalization** In this phase, researchers attempt to make information extracted from retrieved papers comparable. This can be seen as a two-step process: disambiguating extracted information into the intended schema, and normalizing numerical values to be comparable, in both scale and units.
- **Visualization and Modeling** The goal of data-driven design projects is typically not just the extraction and disambiguation of information, but using it to visualize existing literature, in order to plan future experiments, or to serve as a preliminary screen for promising new candidate materials by predicting properties of interest, such as in Zhang et al. (2024).

We argue that a useful evaluation for IE systems is to focus on the middle two phases, namely information extraction and filtering, and disambiguation and normalization. With this scope, we aim to present a system with the content of a paper and the desired schema, and have it output normalized information from the paper in that schema. Systems that perform well at this evaluation would be immensely useful to materials scientsts: given a collected set of potentially useful papers and a desired schema, the system could automate the construction of a dataset that allows visualization and modeling of new materials systems.

2.1 Task Settings

This scope, however, can still present logistical challenges to contemporary approaches: many strong models and systems remain unimodal, and the entire length of a paper may not fit within their context windows. This can be especially challenging when extracting information from the text content of a document, given that it can occur anywhere within a paper. To localize the necessary information, while preserving the challenging, multimodal characteristics of the tasks, we therefore focus on extraction from tables, rather than from text or supplementary information.

With these limitations in mind, we design our task such that the model receives a localized view containing the information to be extracted in one of two formats: either multimodal, in which the model receives a PDF or PNG representation of the page containing the table and related information to be extracted, or text-only, in which the model receives an HTML or XML representation of the

necessary tables, captions, and footnotes. In both of these settings, models are still required to extract information into the provided schema where it exists, ignore irrelevant information and normalize data to fit the schema.

3 Dataset Construction

To demonstrate the challenge of this task scoping, we aim to replicate a subset of the dataset generated from the DDD pipeline developed in Jensen et al. (2019). The original dataset [1] consists of synthesis parameters and derived products of germanium-containing zeolites from 116 papers, resulting in 1638 rows of data. Zeolite synthesis typically involves creating a gel from several components: the elements that form the crystal, such as silicon and germanium, additional reaction components, such as water, and an organic molecule that directs the crystal formation. This dataset contains twelve columns of these ingredients, as well as several more that represent further normalization of their contents, or the results of corroborating simulations. We simplify this dataset, removing information not originally found in the text.

In keeping with the constraints discussed in Section 2, we filter this dataset to information extracted from tables only. The original dataset facilitates this with a column indicating where the data was sourced from in the given article: *Table*, *Text*, or *Supp* (indicating supplementary information). We use only data marked *Table*. We note, however, that the authors of the original study manually reviewed and corrected the extracted information, and in the process, included information that is not originally extracted from tables, and may not be within the one-page context that we provide to our models. For this reason, and for granular error analysis, we additionally annotate the dataset for the location of the data into eight categories falling into three buckets: (1) Data from the table itself (entire columns for that data, information in headers, or information in particular cells under hierarchical indices); (2) Data from related text (table footnotes and captions, or text on the page); or (3) not present (not present within the page context, or not present in the paper at all). We present a visualization of how data are distributed alone these buckets in Figure 2.

Additionally, we omit one paper that contained none of the relevant information to this task on

[1]Available at: https://github.com/olivettigroup/table_extractor/blob/master/zeolite_data/ge_synthesis_data.csv

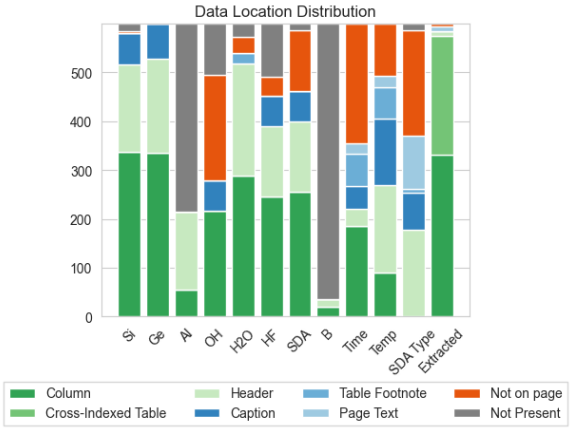

Figure 2: Distribution of data locations in the dataset per column type. Green bars indicate information found within tables, blue indicates related text, and orange/-grey indicates information not available to the models. We note that common information not found on the same page is usually in the paper's synthesis section, which is often not where the synthesis table is found.

the same page as the corresponding table.[2] This results in 28 papers, 601 rows of data, and 7,188 total individual data points. We show a sample of the dataset in Table 1, which corresponds to the reproduced table in Figure 3. We discuss the table's features and work through an example instance of the dataset in sections 3.2 and 3.3, respectively.

3.1 Data Collection and Processing

3.1.1 Multimodal Setting

We manually collected PDFs for all 28 papers, and manually verified the page location of the table from which data was extracted. We then generated both single-page PDF documents and PNG images of that page as input for the multimodal setting. For this setting, we performed no additional post-processing.

3.1.2 Text-only Setting

For the text-only setting, we used publisher APIs where available to download a full-text XML representation of the article's content, and extracted the table, caption, and any other relevant content into a separate document. If an XML representation was not available, we scraped the HTML representation of the paper, and extracted the same elements from the HTML representation in a separate file. In each case, we wrap the content into a top-level tag to

[2]This can arise when e.g. all samples are described and named in a different part of the article, and only those sample names are referenced in the table reporting experimental results/characterization.

make the whole file valid XML/HTML to allow for error-free parsing.

We performed a minimal degree of post-processing on the extracted HTML/XML to allow the content to more easily fit within shorter context lengths; we removed redundant declarations like per-tag XML namespaces. However, we left tag metadata like HTML and CSS classes and styling information in place given their potential semantic utility in table understanding.

For the text-only setting, we do not include any text not directly linked to a table, with the reasoning that searching for the correct information to include would bias our results; position/page information is not as readily available in an XML document. Additionally, in one case, Figure 3 from Corma et al. (2006)[3], the authors use grayscale color fills of cells rather than text in a table to indicate synthesis products. We do not consider this table to have a valid textual interpretation without the necessary resolution of grayscale value to synthesis product name. As a result, we omit this table from the textual setting of this benchmark, resulting in 457 rows of data.

Table 1
Selection of the most representative synthesis of zeolite IM-16 with 3-ethyl-1-methyl-3*H*-imidazol-1-ium as OSDA.

Sample	Molar gel composition (T = Si+Ge)				Material
	H_2O/T	R/T	HF/T	Si/Ge	
1[a]	20	0.5	0	0.6:0.4	**TON**+**MFI**+Arg[c]
2[a]	20	1	0	0.6:0.4	**MFI**+ε?[d]
3[a]	8	0.5	0.5	1:0	Amorphous
4[a]	8	0.5	0.5	0.8:0.2	IM−16+ε?[d]
5[a]	8	0.5	0.5	0.6:0.4	IM−16+ε?[d]
6[a]	8	0.5	0.5	0.4:0.6	Q[c]+IM−16
7[a]	8	0.5	0.5	0.2:0.8	Q[c]
8[a]	8	0.6	0.4	0.8:0.2	IM−16+ε?[d]
9[a]	20	1	0.5	0.6:0.4	IM−16+ε?[d]
10[a]	3	0.3	0.3	0.8:0.2	IM−16+ε?[d]
11[a]	20	1	1	0.8:0.2	IM−16+ε?[d]
12[a]	20	1	1	0.6:0.4	IM−16+ε?[d]
13[a]	20	1	1	0.5:0.5	IM−16+Q[e]
14[b]	20	1	1	0.8:0.2	IM−16+**MFI**
15[b]	20	1	1	0.6:0.4	IM−16+ε?[d]
16[b]	20	1	1	0.5:0.5	IM−16

Silica sources:
[a] TEOS (tetraethylorthosilicate).
[b] Aerosil 200.
[c] Argutite.
[d] ε?: small quantity of one or more unknown impurities.
[e] Quartz.

Figure 3: Example table from the dataset, reproduced from Lorgouilloux et al. (2009, Table 1). This table demonstrates several of the challenges with table extraction in this dataset, including: (1) Generic table layout understanding; (2) Processing information related to tables, such as captions and footnotes; (3) Understanding and resolving in-document substitutions; and (4) Numerical reasoning to normalize ratios.

3.2 Task Features

This task presents a number of interesting challenges to information extraction methods. In this section, we discuss these features, using a representative table from Lorgouilloux et al. (2009, Figure 3).

Table Understanding. The core challenge of this task is processing tables in the variety of forms in which they occur. Tables expressing synthesis parameters and recipes are difficult to construct: Experiments often involve the systematic variation of several different parameters, leading to a challenge in how to represent hierarchical data in many dimensions in an ultimately two-dimensional table. This results in a number of different formats. Figure 3 demonstrates perhaps the most common format, normalized rows per-experiment, but hierarchical representations that involve leaving cells blank to indicate a hierarchical grouping of experiments are also common, and pose a challenge for table understanding models.

Related Information. While this task is primarily oriented around table extraction, information necessary to understanding the table is frequently presented around the table. Figure 3 specifies the OSDA compound in the caption, and additionally specifies the expansion of several acronyms in table footnotes, which are placed directly below the table. Further, many papers introduce information necessary for table understanding in the text surrounding the tables, leading to our benchmark setting providing the full page context for a table. We note that table captions can be an edge case for some approaches: The VILA (Shen et al., 2022a) model, for example, detects the table caption as part of the table, which can lead some table understanding models to parse table captions as further rows of the table, rather than footnotes. Further, understanding non-table information here requires the resolution of superscripts to their corresponding footnotes.

Numerical Reasoning. Synthesis procedures for zeolites are commonly expressed in terms of molar ratios of the components, and the choice of which element to which to normalize changes interpretation of numerical values in the table. For example, in Figure 3, ratios are scaled to the combination of silicon and germanium in the sample. By contrast, several other papers (and the final dataset) scale to only the quantity of silicon, requiring a normaliza-

[3]DOI: 10.1016/j.jcat.2006.04.036

Si	Ge	Al	OH	H_2O	HF	SDA	B	Time	Temp	SDA Type	Extracted
1	0.667	0	0.8335	33.34	0	0.8335	0	336	170	3-ethyl-1-meth...	TON+MFI+argutite
1	0.667	0	1.667	33.34	0	1.667	0	336	170	3-ethyl-1-meth...	MFI+unknown
1	0	0	0.5	8	0.5	0.5	0	336	170	3-ethyl-1-meth...	Amorphous
1	0.25	0	0.625	10	0.625	0.625	0	336	170	3-ethyl-1-meth...	IM-16+unknown

Table 1: Sample rows from our dataset, filtered from Jensen et al. (2019). This table represents the first four rows of the table seein in Figure 3. For space, we omit the columns where we describe where the data was located.

tion step that introduces a multiplicative factor to enable direct comparison of results across papers.

Within-document Reference Resolution. Before normalization can occur, tables often require the resolutions of symbols that are defined elsewhere in the document. In this case, the table headers indicate that the H_2O, R, and HF columns are normalized to T, which the upper header declares as the combination of silicon and germanium in the sample.

Sparsity. In many cases in the extracted dataset, columns will have values of 0, because a given element was not used. Systems that attempt this benchmark must not hallucinate non-zero values even when given a comprehensive schema of all items that may or may not be present.

3.3 A Worked Example

Figure 3 represents indices 375-390 from our dataset. We reproduce the first four rows of this table here, and demonstrate how to extract the relevant columns in the first row.

If present, the silicon content is always the basis of normalization, and so receives a value of 1 in the `Si` column. This therefore leads us to normalize the germanium value, in the ratio of Si:Ge 0.4:0.6, to 0.667. This paper uses neither aluminum nor boron, leading to 0 values for both of those. Water and HF content are similarly normalized by dividing by 0.6.

In the table in Figure 3, the R column is interpreted as the OSDA, even though this is not specified in the paper. This is a common substitution, alongside others, such as using "T" as the basis for normalization. We therefore use the values in the R column for the SDA value.

Text found elsewhere on the page provides additional information that must be incorporated. Synthesis paragraph 2.1 implies that the OSDA is also the source of OH ions: "and 3-ethyl-1-methyl-3H-imidazol-1-ium bromide (98%, Solvionic), which was transformed into its OH form by ion exchange in water." The time and temperature (170°C for 14 days) are from the same paragraph; 14 days must be normalized to 336 hours.

The name of the OSDA is specified in the table caption. The names of the products are extracted into column S, but must be expanded using the table footnotes to indicate that "Arg" is argutite, and "Q" is quartz.

This table demonstrates several of the challenges in this dataset, from table understanding, to resolving in-table references, having conventional knowledge, and using contextual text that is not explicitly part of the table being considered or extracted.

3.4 Evaluation

Given the information extraction-based nature of this task, we consider an F1 metric, with some allowance for what is counted as a match. In the case of numeric columns, to allow for imprecision in normalization of ratios, we consider a "correct" answer to be within 5 of the true answer. In the case of the OSDA name column and the extracted products column, we expect an exact match on a lowercased version of the string with all punctuation replaced by an underscore; in the case of the extracted products column, we note that often, several products of a reaction are mentioned; we intend to improve the granularity of our evaluation in ongoing work. Given the large variance of the number of rows/data points extracted from individual articles, we consider a micro-averaged F1 score to be an appropriate choice.

For evaluation, we provide code that accepts a spreadsheet with the same header row as the original dataset (omitting the location rows). This code expects ten numeric and two text columns. `None` values indicate that the model is deliberately not providing a response, to disambiguate from cases where the correct extraction is zero, or another common placeholder value. For our F1 metric, we consider any data that is available on the page (i.e. not annotated as being "not on page" or "not present") a candidate for extraction. A true positive is any

data point that is available to the model and correctly extracted; false negatives are any point that the model fails to extract. False positives include both incorrectly extracted values and values that are not available to the model, but that it provided a value for anyway. True negatives are information not available to the model that it successfully does not provide a value for. We micro-average the F1 across papers, and additionally provide per-location F1 scores to indicate what sources of information models are adept at working with.

However, evaluation does pose additional challenges: While some tables translate straightforwardly between rows in the original table and the dataset, others are structured differently, using hierarchical indices, such that blank cells' content must be inferred, or tables with multiple levels of hierarchy, where one cell and the headers that index it correspond to a row in the final dataset. We call these tables *cross-indexed*. Further, while the table reproduced in Figure 3 uses identifiers for individual samples, that is not common in our dataset. As a result, there is no *a priori* alignment between rows in the dataset and rows produced from models solving this task.

To address this, we use a simple heuristic algorithm that attempts to align rows in the dataset with rows produced from the systems under evaluation, with strong priors towards the initial alignment being correct. Our algorithm begins by computing a row-wise score between all rows in the dataset and predictions. This score computes the match discussed above on all columns where information is within the provided context window, to avoid spurious matches on absent information. We then iterate through each row of the dataset, and choose the highest scoring predicted row to align with each row in the dataset. In the case of a score tie, the sequentially following row is assigned. Because of the varying structures of tables in the dataset, we additionally implement fallbacks in the case of a mismatched number of rows between the dataset and predictions. In the case where the model produces more rows than are observed in the dataset, each additional row is penalized as being false positives; in the case where the model produces too few rows, we construct placeholder rows of no predictions to indicate that the model has not provided an answer. We note both that this alignment strategy is not guaranteed to produce the optimal alignment, but also that any similar strategy will end up favoring models by potentially offering mistaken credit.

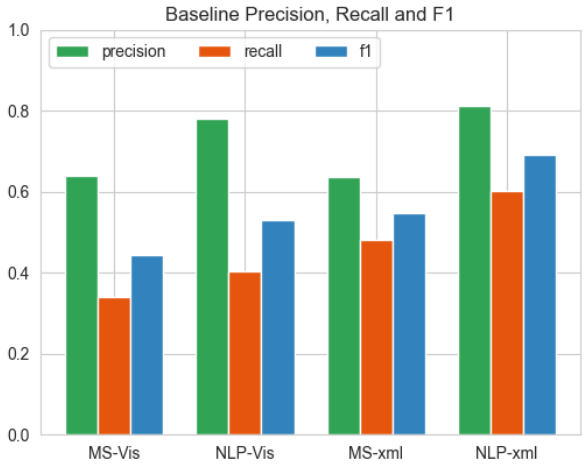

Figure 4: Baseline results. We visualize all four baseline settings here, and note that the NLP-xml setting gained a notable advantage over other settings.

We plan to iterate on this in future versions of this work.

4 Baselines

Despite recent work investigating Large Language Models (LLMs) as possible automated scientists (Lu et al., 2024; Si et al., 2024), to our knowledge LLMs have never been systematically evaluated on research processes such as precise multi-document review and synthesis.

As a baseline, we evaluated a prompt-based strategy with a multimodal Large Language Model, `GPT-4o` (OpenAI et al., 2024). Our goal was for the model to perform both the information extraction and table normalization jointly when provided with either an image (300 dpi PNG) of the PDF page, or the raw underlying XML of the document.

We selected two prompt constructors from among the authors of this paper: One, a graduate student in NLP; the other, a graduate student in materials science. This setup gives us insight into the process of prompt construction coming from either NLP or materials science expertise. Each constructor was provided with the same three randomly selected articles from the dataset to act as a guide while developing their prompts. We intentionally restricted the prompt constructors' access to the full set of papers so that information and edge cases from the test set would not influence prompt design. The prompt constructors also had a 30-minute discussion about prompting strategy, but otherwise constructed their prompts independently of one another.

After receiving prompts from the annotators we

made minor edits to produce consistent JSON structured final output, and to make the prompts suitable for both image and XML inputs. Outputs were then post-processed so that column names aligned with the evaluation data.

Each prompt was applied to the full test dataset of papers, first with pages represented as images and then in XML form. This gave us four baseline outputs: `MS-Vis` (materials scientist prompter, for visual modality), `MS-xml` (materials scientist prompter, for XML modality), `NLP-Vis` (NLP prompter, for visual modality), and `NLP-xml` (NLP prompter, for xml modality). The full text of the prompts is included in Appendix A.

5 Results and Discussion

We summarize our high-level results in Figure 4. Overall, we find that while GPT-4o performed well on this benchmark, there is a long way to go before DDD can be performed fully unsupervised. In particular, we note that in the visual setting, our highest F1 score is 0.54, which rises to 0.69 in the XML setting. But even with access to raw text in the XML setting, precision is well below 1.0, meaning that researchers might still be required to manually verify extracted values. The result that the text-only setting performs better is not surprising, and does point to the promise of DDD via publisher APIs, rather than through scraping PDFs.

In addition, we plot the F1 results against the location from which the data was extracted in Figure 5. This table offers several insights into how the model fares on this benchmark. Perhaps most notably, there is a large difference in performance between different table layouts: Whereas the models are relatively successful at extracting tables where one row in the table maps to one row in the dataset, reaching a top F1 of 0.803, that performance does not generalize to cross-indexed tables where recipe parameter values are placed in hierarchical column and row indices, and table cell values indicate the synthesis recipe corresponding to those values.

We also see differences across modalities. Table headers are much more successfully parsed in the text-only setting, across both of our prompters, where table captions are better parsed in a visual setting. Interestingly, footnotes seem to be the most robustly parsed type of data, outperforming even conventional table extraction.

6 Open Questions and Future Work

We present in this paper a preliminary version of the dataset that we hope to iterate on. In this section, we discuss ongoing work and future directions for this benchmark. We'd appreciate discussion on these points!

More Thorough Baselines. In this paper, we run four baselines, but atop a single model. In future work, we plan to expand the thoroughness of our baselines by using different models and including in-context learning as a setting.

Grounding. In creating this dataset, we chose to implement an entirely end-to-end evaluation framework. In part, this was designed to test the degree to which contemporary models and systems can be used as a drop-in addition to existing DDD workflows. This is dissimilar, however, from the way that datasets such as FinQA (Chen et al., 2022) are constructed, in which each piece of extracted information is grounded to a place in the text. To what degree is grounding necessary for the measurements of this benchmark to be reliable?

Whole PDF Extraction. In the construction of this benchmark, we deliberately avoided using whole PDFs as a concession to the practicality of processing. This had cascading effects, in that we were then constrained to data that remained localized to a reasonable context (in our case, a page). Given, however, the fairly strong performance that our baselines achieved on this task, designing a whole-PDF/XML version of this benchmark is on the road map for this work.

Scale. In this dataset, we examined a subset of one DDD paper. However, the utility of a zero- or few-shot model for this task is not for its ability to replicate one study, but to be a generalizable tool for DDD more broadly. In future work, we plan to expand along two axes: First, by expanding the scope include both larger studies, and more of the study each time. In particular, Pan et al. (2024) follow up on Jensen et al. (2019)'s work by significantly expanding the scope, while still ensuring quality by manual verification. Second, we plan to expand to another domain, including data on composition-property relationships of aluminum alloys (Pfeiffer et al., 2022), to test the degree to which techniques can transfer across subfields in materials science.

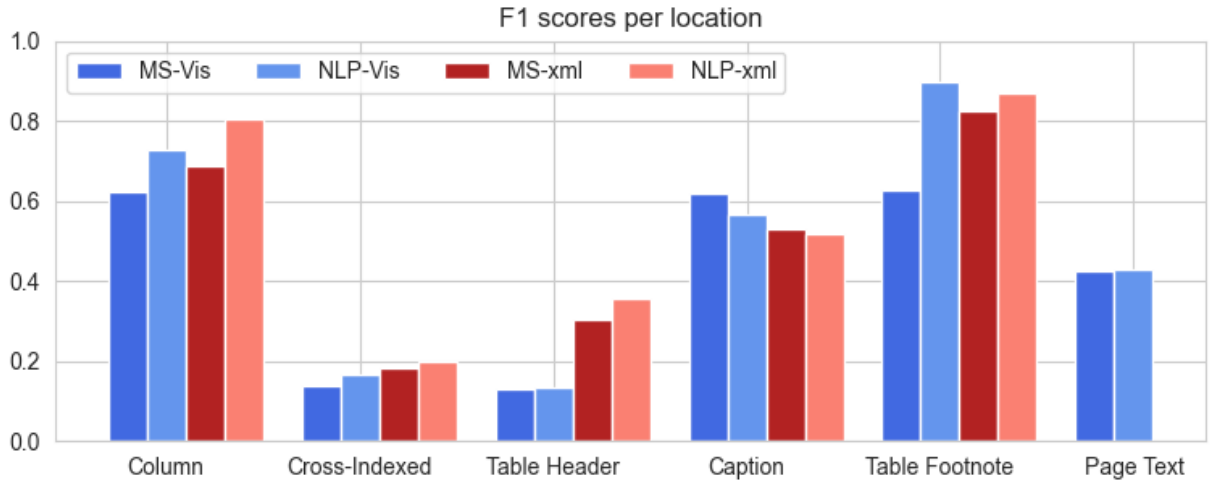

Figure 5: Location-based results. We see a clear differentiation in performance across table extraction from different table types: Cross-indexed tables are difficult in both settings, for both prompters. Notably, we see large differences in the parsing of table headers and captions across settings. XML settings are excluded for the “page text" category, because we did not include paper text in the XML setting, as discussed in section 3.

7 Related Work

Visually Rich Document Understanding The proposed benchmark bears many similarities to work in visually rich document understanding (VRDU). Tasks in VRDU, including to answer questions based on financial documents (Chen et al., 2022), or to understand forms (Jaume et al., 2019), receipts (Park et al.), or to perform information extraction on non-disclosure agreements and financial statements (Stanisławek et al., 2021). Each of these datasets emphasizes the use of visual document features as necessary information to understand the documents' contents. However, while many of these tasks focus on documents that have been scanned and had OCR applied to them, we focus in this paper on scientific documents that are natively digital. We note, however, that scientific literature from before digital typesetting remains of significant interest in many fields.

Scientific Document Understanding Separately from document understanding tasks more generally, work on understanding scientific documents is a growing field. Work like VILA (Shen et al., 2022b) implements document structure recognition on scientific publications, and DDD has historically relied on information extraction tools like ChemDataExtractor (Swain and Cole, 2016; Mavracic et al., 2021) and MatSciBERT (Gupta et al., 2022).

8 Conclusion

In this paper, we repurpose a tabular dataset for data-driven design in materials science as a benchmark for multimodal information extraction. We scope the problem down to page-level information extraction tasks, in which models are expected to pull from a variety of information contexts to satisfy the goal, and expose two settings, multimodal and text-only. In evaluating a model on our benchmark, we see that while models perform well on certain kinds of standardized tables, their performance drops significantly on tables with different layouts, or where information needs to be found elsewhere in the document. We argue that benchmarks oriented towards data-driven design should be strong candidates on which to focus effort in information extraction, both to advance the state of the art in NLP, and for the utility to materials science.

Limitations

We discuss the limitations of this work in several sections of the paper. To summarize here, this work is currently limited both by its scope and granular details of implementation. We hope to address both of these in ongoing work.

References

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and

William Yang Wang. 2022. FinQA: A Dataset of Numerical Reasoning over Financial Data. *arXiv preprint*. ArXiv:2109.00122 [cs].

Avelino Corma, M José Díaz-Cabanas, Manuel Moliner, and Cristina Martínez. 2006. Discovery of a new catalytically active and selective zeolite (ITQ-30) by high-throughput synthesis techniques. *Journal of Catalysis*, 241(2):312–318.

Callum J. Court and Jacqueline M. Cole. 2018. Auto-generated materials database of Curie and Néel temperatures via semi-supervised relationship extraction. *Scientific Data*, 5(1):180111. Publisher: Nature Publishing Group.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418. Publisher: Nature Publishing Group.

Tanishq Gupta, Mohd Zaki, NM Anoop Krishnan, and Mausam. 2022. Matscibert: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6.

Zach Jensen, Edward Kim, Soonhyoung Kwon, Terry Z. H. Gani, Yuriy Román-Leshkov, Manuel Moliner, Avelino Corma, and Elsa Olivetti. 2019. A Machine Learning Approach to Zeolite Synthesis Enabled by Automatic Literature Data Extraction. *ACS Central Science*, 5(5):892–899. Publisher: American Chemical Society.

Edward Kim, Kevin Huang, Alex Tomala, Sara Matthews, Emma Strubell, Adam Saunders, Andrew McCallum, and Elsa Olivetti. 2017. Machine-learned and codified synthesis parameters of oxide materials. *Scientific Data*, 4(1):170127. Publisher: Nature Publishing Group.

Yannick Lorgouilloux, Mathias Dodin, Jean-Louis Paillaud, Philippe Caullet, Laure Michelin, Ludovic Josien, Ovidiu Ersen, and Nicolas Bats. 2009. IM-16: A new microporous germanosilicate with a novel framework topology containing *d4r* and *mtw* composite building units. *Journal of Solid State Chemistry*, 182(3):622–629.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *Preprint*, arXiv:2408.06292.

Juraj Mavracic, Callum J Court, Taketomo Isazawa, Stephen R Elliott, and Jacqueline M Cole. 2021. Chemdataextractor 2.0: Autopopulated ontologies for materials science. *Journal of Chemical Information and Modeling*, 61(9):4280–4289.

Santiago Miret and N. M. Anoop Krishnan. 2024. Are LLMs Ready for Real-World Materials Discovery? *arXiv preprint*. ArXiv:2402.05200 [cond-mat].

Elsa A. Olivetti, Jacqueline M. Cole, Edward Kim, Olga Kononova, Gerbrand Ceder, Thomas Yong-Jin Han, and Anna M. Hiszpanski. 2020. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews*, 7(4):041317.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider,

Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Elton Pan, Soonhyoung Kwon, Zach Jensen, Mingrou Xie, Rafael Gómez-Bombarelli, Manuel Moliner, Yuriy Román-Leshkov, and Elsa Olivetti. 2024. ZeoSyn: A Comprehensive Zeolite Synthesis Dataset Enabling Machine-Learning Rationalization of Hydrothermal Parameters. *ACS Central Science*, 10(3):729–743.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing.

Olivia P. Pfeiffer, Haihao Liu, Luca Montanelli, Marat I. Latypov, Fatih G. Sen, Vishwanath Hegadekatte, Elsa A. Olivetti, and Eric R. Homer. 2022. Aluminum alloy compositions and properties extracted from a corpus of scientific manuscripts and us patents. *Scientific Data*, 9(128).

Zejiang Shen, Kyle Lo, Lucy Lu Wang, Bailey Kuehl, Daniel S. Weld, and Doug Downey. 2022a. VILA: Improving structured content extraction from scientific PDFs using visual layout groups. *Transactions of the Association for Computational Linguistics*, 10:376–392.

Zejiang Shen, Kyle Lo, Lucy Lu Wang, Bailey Kuehl, Daniel S Weld, and Doug Downey. 2022b. Vila: Improving structured content extraction from scientific pdfs using visual layout groups. *Transactions of the Association for Computational Linguistics*, 10:376–392.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *Preprint*, arXiv:2409.04109.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. Kleister: Key Information Extraction Datasets Involving Long Documents with Complex Layouts. volume 12821, pages 564–579. ArXiv:2105.05796 [cs].

Matthew C. Swain and Jacqueline M. Cole. 2016. ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. *Journal of Chemical Information and Modeling*, 56(10):1894–1904. Publisher: American Chemical Society.

Hengrui Zhang, Alexandru B. Georgescu, Suraj Yerramilli, Christopher Karpovich, Daniel W. Apley, Elsa A. Olivetti, James M. Rondinelli, and Wei Chen. 2024. Emerging Microelectronic Materials by Design: Navigating Combinatorial Design Space with Scarce and Dispersed Data. *arXiv preprint*. ArXiv:2412.17283 [cond-mat].

Zhiling Zheng, Oufan Zhang, Christian Borgs, Jennifer T. Chayes, and Omar M. Yaghi. 2023. ChatGPT Chemistry Assistant for Text Mining and the Prediction of MOF Synthesis. *Journal of the American Chemical Society*, 145(32):18048–18062. Publisher: American Chemical Society.

A Prompts

We provide the prompts used by both prompt writers in this section below.

A.1 MS-Vis

```
\
You are a journal analyst, expert in data
    extraction and analysis. \
You will be provided with a image as a page
    captured from a journal. \
Your task is to extract the synthetic recipe
    information contained in the table and
    format it into JSON for every single entry.\
Show full response for every step.\
Between every step, think again out loud if the
    adjustments are correct or not, if not, make
     adjustments and evaluate until you think it
     is correct.
The final JSON recipe should contain
1. all the molar fraction(atomic percentage) of
    every metal element and common chemical
    compounds
2. the condition of that synthetic process such
    as time and temperature
3. (optional) additional agents used in the
    synthesize process.
Here are the steps:
Step 1: Extract data from the table itself and
    reconstruct it as a markdown table using
    latex expression for all entry. Pay extra
    attention on the superscript and subscripts,
     skip the ones that are footnotes. If values
     presented contain a range, use the mean of
    the range.
Step 2: Read the caption and all the text that
    strongly relates to this table
Step 3: Adjust or fill-in the table based on the
     information gathered in step 2. Replace the
     specific abbreviation only used in this
    journal with its well defined name. If the
    abbreviation is widely used, don't replace
    it. If you are not sure about the
    abbreviation, leave it as its original form.
Step 4: Reconstruct the table to reduce the
    dimension. Every row would be a single
    synthesis process recipe. Keep the format
    same as markdown table with latex expression
    .
Step 5: Replace the ratios with the recipe's
    molar fraction for every metal element and
    common chemical compounds. For recipes
    containing Si, normalize Si to 1. For
    recipes containing Ge, normalize Ge to 1 if
    Si is absent.
Step 6: Convert the markdown table into JSON
    format, each recipe should be similar to the
     structure below
{recipe_format}

Your final response should be a JSON object of
    the following form:

{{
   "step1": <>,
   "step2": <>,
   "step3": <>,
   "step4": <>,
   "step5": <>,
   "recipes": [
      {recipe_format},
      {recipe_format},
      ...,
      ]
}}
```

A.2 MS-XML

```
   \
XML Document:

{context}

You are a journal analyst, expert in data
    extraction and analysis. \
You have been given an XML document of a paper
    captured from a journal. \
Your task is to extract the synthetic recipe
    information contained in the table and
    format it into JSON for every single entry.\
Show full response for every step.\
Between every step, think again out loud if the
    adjustments are correct or not, if not, make
     adjustments and evaluate until you think it
     is correct.
The final JSON recipe should contain
1. all the molar fraction(atomic percentage) of
    every metal element and common chemical
    compounds
2. the condition of that synthetic process such
    as time and temperature
3. (optional) additional agents used in the
    synthesize process.
Here are the steps:
Step 1: Extract data from the table itself and
    reconstruct it as a markdown table using
    latex expression for all entry. Pay extra
    attention on the superscript and subscripts,
     skip the ones that are footnotes. If values
     presented contain a range, use the mean of
    the range.
Step 2: Read the caption and all the text that
    strongly relates to this table
Step 3: Adjust or fill-in the table based on the
     information gathered in step 2. Replace the
     specific abbreviation only used in this
    journal with its well defined name. If the
    abbreviation is widely used, don't replace
    it. If you are not sure about the
    abbreviation, leave it as its original form.
Step 4: Reconstruct the table to reduce the
    dimension. Every row would be a single
```

```
    synthesis process recipe. Keep the format
    same as markdown table with latex expression
    .
Step 5: Replace the ratios with the recipe's
    molar fraction for every metal element and
    common chemical compounds. For recipes
    containing Si, normalize Si to 1. For
    recipes containing Ge, normalize Ge to 1 if
    Si is absent.
Step 6: Convert the markdown table into JSON
    format, each recipe should be similar to the
     structure below
{recipe_format}

Your final response should be a JSON object of
    the following form:

{{
   "step1": <>,
   "step2": <>,
   "step3": <>,
   "step4": <>,
   "step5": <>,
   "recipes": [
      {recipe_format},
      {recipe_format},
      ...,
      ]
}}
```

A.3 NLP-Vis

```
   \
You are a materials science research assistant
    agent. Your task is to visually analyze
    papers from the materials science field and
    extract information about {recipe_type}
    recipes.
Each image will contain a table describing the
    synthesis recipe. This table will contain
    information about the recipe including:
- Ratios of the reaction reagents, including {
    reagents} and other elements
- Information on the temperature and duration of
     the reaction
- The structure directing agent, which guides
    the formation of the zeolites

You must perform the following steps, using your
     own visual capabilities (which are
    significant and have been highly improved by
     {model_makers}) and not relying on external
     tools.
1. Read the contents of the table, and duplicate
     that table as a csv within your response.
    Make sure to carefully read possible
    subscript (for instance, for element ratios
    in a molecular formula) from the table, and
    distinguish them from footnotes in the table
    . Do not abbreviate the table. If values
    presented contain a range, use the mean of
    the range.
2. Identify which properties from the property
    list below are included in the table. Many
    of the properties relate to quantities or
    ratios of reagents. Sometimes the column
    will be named based on the source material (
    ie SiO2 for Si). Treat those as the
    corresponding element. Write out a mapping
    between columns and properties.
3. Read the text to find other recipe
    information that are not contained in the
    table text, but may be mentioned in the text
     of the paper or the caption of the table.
4. Expand any abbreviations used in the recipe.
    For instance, if the recipe describes U = Na
     + Cl, that means that "U" represents the
    total amount of Na and Cl in the recipe.
    Write the expanded abbreviations below the
    table and other relevant information.
5. Determine the ratio for each reagent. Setting
     Silicon to "1", determine the proportion
    for each reagent relative to the silicon.
    Sometimes the table will already do this; in
     that case, replicate it from the table. But
     if a Si/Ge ratio of .5 is described, Si = 1
     and Ge = 2. You can write out the
    mathematical expressions used to perform
    these calculations.
6. Rewrite the csv table as a JSON containing
    adjusted values. For instance, if "U" ( = Na
     + Cl) had its own column, create a column
    for Na and a column for Cl. The resulting
    recipe list must be a list of JSON objects,
    with each object corresponding to one recipe
     and its properties.

Ignore information related to the resulting
    properties of the resulting compound, only
    focus on the parameters/instructions used to
     perform the recipe. If an expected value in
     the recipe (listed below), fill that value
    with the empty string.

The properties of interest are:
{properties}
Do not include any properties except for these (
    or properties which are equivalent)!

The JSON response should be in this format:

{{
   "table csv": <open text>,
   "other information": <open text>,
   "property_mapping": <open text>,
   "formula abbreviations": <open text>,
   "ratio calculations": <open text>,
   "recipes": [
      {recipe_format},
      {recipe_format},
      ...,
      ]
```

A.4 NLP-XML

```
   \
XML Document:

{context}

You are a materials science research assistant
    agent. Your task is to analyze papers from
    the materials science field and extract
    information about {recipe_type} recipes.
You have been given an XML document containing a
     table describing the synthesis recipe. This
     table will contain information about the
    recipe including:
- Ratios of the reaction reagents, including {
    reagents} and other elements
```

```
- Information on the temperature and duration of
     the reaction
- The structure directing agent, which guides
    the formation of the zeolites

You must perform the following steps, using your
     own reasoning capabilities (which are
    significant and have been highly improved by
     {model_makers}) and not relying on external
     tools.
1. Read the contents of the table, and duplicate
     that table as a csv within your response.
    Make sure to carefully read possible
    subscript (for instance, for element ratios
    in a molecular formula) from the table, and
    distinguish them from footnotes in the table
    . Do not abbreviate the table. If values
    presented contain a range, use the mean of
    the range.
2. Identify which properties from the property
    list below are included in the table. Many
    of the properties relate to quantities or
    ratios of reagents. Sometimes the column
    will be named based on the source material (
    ie SiO2 for Si). Treat those as the
    corresponding element. Write out a mapping
    between columns and properties.
3. Read the text beyond the table to find other
    recipe information that are not contained in
     the table text, but may be mentioned in the
     text of the paper or the caption of the
    table.
4. Expand any abbreviations used in the recipe.
    For instance, if the recipe describes U = Na
     + Cl, that means that "U" represents the
    total amount of Na and Cl in the recipe.
    Write the expanded abbreviations below the
    table and other relevant information.
5. Determine the ratio for each reagent. Setting
     Silicon to "1", determine the proportion
    for each reagent relative to the silicon.
    Sometimes the table will already do this; in
     that case, replicate it from the table. But
     if a Si/Ge ratio of .5 is described, Si = 1
     and Ge = 2. You can write out the
    mathematical expressions used to perform
    these calculations.
6. Rewrite the csv table as a JSON containing
    adjusted values. For instance, if "U" ( = Na
     + Cl) had its own column, create a column
    for Na and a column for Cl. The resulting
    recipe list must be a list of JSON objects,
    with each object corresponding to one recipe
     and its properties.

Ignore information related to the resulting
    properties of the resulting compound, only
    focus on the parameters/instructions used to
     perform the recipe. If an expected value in
     the recipe (listed below), fill that value
    with the empty string.

The properties of interest are:
{properties}
Do not include any properties except for these (
    or properties which are equivalent)!

The JSON response should be in this format:

{{
    "table csv": <open text>,
    "other information": <open text>,
    "property_mapping": <open text>,
    "formula abbreviations": <open text>,
    "ratio calculations": <open text>,
    "recipes": [
        {recipe_format},
        {recipe_format},
        ...,
        ]
}}
```