Using machine-learning and large-language-model extracted data to predict copolymerizations

Mara Schilling-Wilhelmi [1] **Kevin Maik Jablonka** [1 2 3]

Abstract

Predicting the outcome of chemical reactions using machine learning (ML) approaches can significantly enhance research in chemistry and materials science. The synthesis of polymers, for instance, depends heavily on reaction conditions such as temperature and solvent, making it challenging to predict products with only monomer information. In this work, we address this challenge by compiling the first comprehensive copolymerization dataset, including reaction conditions, consisting of 1138 reactions involving 347 unique monomers. We employed vision language models (VLMs) to extract data from 361 scientific articles, overcoming the limitations of traditional visual document understanding tools. In addition, we developed a novel data-driven filtering approach to further improve performance. Using this data, we built the first predictive models for copolymer reactivity that can predict whether a given reaction system favors homopolymerization. Our work showcases how advances in machine learning, in particular Large Language Models (LLMs), make it possible to address complex problems by creating bespoke datasets in a very flexible and scalable fashion.

1. Introduction

Modeling and predicting chemical reactions using machine-learning approaches can greatly accelerate chemistry and material science research. For instance, machine learning (ML) algorithms could optimize reaction conditions or provide reactants for a desired product by learning from historical data or even "failed" reactions (Moosavi et al., 2019; Raccuglia et al., 2016; Shields et al., 2021). While some success has been achieved in modeling the outcome of chemical reactions based on only the reactants (Coley et al., 2019; Schwaller et al., 2019; Segler et al., 2018), the success of chemical reactions often depends on the choice of the right reaction conditions, such as temperature or solvent (Gao et al., 2018; Jorner et al., 2021). This applies especially to polymer science, as one pair of monomers could react (or not) in a practically infinite number of ways. For instance, the same monomers can form chains of different lengths or chains with different patterns of monomer linkages (random, blocks, etc.). To describe the behavior of copolymerization reactions, reaction parameters such as r-values and Q,e-values have been developed. For instance, in the framework of r values, there is one value per polymer for a given reaction, and knowledge of the r values lets one determine the resulting polymer architecture (e.g., $r_1 = r_2 = 1$ describes ideal statistical copolymerization) (Cowie & Arrighi, 2007). Importantly, those reaction parameters do not only depend on the reactants but crucially on the choice of reaction conditions (Lewis et al., 1948). Therefore, performing a practically useful prediction of polymerization requires a data set that includes not only reactants and products but also reaction conditions. Such a dataset currently does not exist and is also difficult to obtain, as the information is spread across tables and text in dated papers, for which often only scans are available Figure 5.

In this work, we compile the first comprehensive copolymerization dataset by extracting data using vision language models (VLMs) and use it to build a model that can predict limiting cases of copolymerization reactions.

Concretely, our main contributions are:

- A novel machine-learning-ready dataset of copolymerization reactions of 1138 reactions, spanning 347 unique monomers along with information about reaction conditions.
- A detailed analysis of the performance of various extraction pipelines, spanning ML-based visual document understanding (VDU)-tools as well as multi-

[1]Laboratory of Organic and Macromolecular Chemistry (IOMC), Friedrich Schiller University Jena, Humboldstr. 10, 07743 Jena, Germany [2]Center for Energy and Environmental Chemistry Jena, Friedrich Schiller University Jena, Philosophenweg 7, 07743 Jena, Germany [3]Helmholtz Institute for Polymers in Energy Applications Jena (HIPOLE Jena), Lessingstraße 12-14, 07743, Jena, Germany. Correspondence to: Mara Schilling-Wilhelmi <mara.wilhelmi@uni-jena.de>, Kevin Maik Jablonka <mail@kjablonka.com>.

modal LLMs.

- A model-based filtering approach to increase the precision of the extraction pipeline from $77\,\%$ to $94\,\%$.
- A machine learning model to important limiting cases of copolymerization behavior based on information on reactants and reaction conditions.

Overall, our work will enable the systematic tuning of copolymerization conditions to tailor polymers to desired chain compositions and lengths.

2. Related work

Structured data extraction using language models Structured data extraction in chemical and materials science has a long history. For a long time, hard-coded extraction rules such as regular expressions have been performed, which have limited the applicability and transferability of bespoke extraction pipelines (Hawizy et al., 2011; Swain & Cole, 2016). With the recent advances in generally applicable LLMs, success has been achieved by simply prompting (Jablonka et al., 2023; Patiny & Godin, 2023; Polak & Morgan, 2024) or fine-tuning those general-purpose models for scientific data extraction (Ai et al., 2024; Dagdelen et al., 2024).

Modeling of copolymerization reactions While there has been some advance in modeling reactivity ratios using techniques such as density functional theory (DFT) (Dossi & Moscatelli, 2012; Filley et al., 2002), those techniques are still too expensive and typically bespoke for particular systems to be useful for routine optimization of reaction conditions. While there have been initial attempts in utilizing ML to predict reactivity ratios, those only have limited applicability as they fail to model the impact of reaction conditions (Farajzadehahary et al., 2023; Fazakas-Anca et al., 2021; Nguyen & Bavarian, 2023).

3. Results and discussion

3.1. Structured data extraction pipeline

VDU followed by LLMs The most common way to utilize PDF documents in a language modeling framework is to convert them into structured text data using visual document understanding techniques, which can also include Optical Character Recognition (OCR). For instance, one can convert the PDF into a machine-readable format using a ML-based VDU tool like Nougat (Blecher et al., 2023) or Marker (Paruchuri, 2023) and extract the data by using a general-purpose LLMs like GPT-3.5 Turbo. However, these VDU-tools tend to struggle to convert complex structures like tables and cannot utilize data in plots. Therefore, such information might be lost before the LLMs could extract the data.

This is also what we observe in our experiments, where Figure 2 shows that the pipelines that leverage VDU tools typically show only low, not practically useful, precision. The manual analysis of the errors revealed that those VDU tools have a high error rate in converting tables. We observe with the Nougat tool that only $45\,\%$ of the tables are converted, of which $44\,\%$ are converted correctly. As a result, only $20\,\%$ of the tables in the articles are available and correct for data extraction. This presents a performance bottleneck for our application since most of the relevant data is reported in tables.

For this reason, we also explored other approaches such as "Assistants" (i.e., tool-augmented LLMs). While the GPT-4 Assistant tool can utilize various possible input formats, including PDF, which seems like a very convenient solution, it has limited applicability in a scientific setting as one has no insight into how the files get processed. In addition, the very high cost of GPT-4 Assistant even further limits its applicability for a high-throughput data-extraction setting.

Vision-language models A common pattern for improving ML systems is to use end-to-end learning, which avoids intermediate processing steps but instead models the entire problem end-to-end. In our case, models that can process images beside the text, so-called VLMs, can be used to avoid the conversion steps (such as VDU) in between. The most performant VLMs are GPT-4 Vision, GPT-4o, and Claude 3 Opus (Center for Research on Foundation Models, 2024). To use any of them, PDFs need to be converted into images, for which one can choose different target resolutions. In our experiments, this image resolution seems to be the limiting factor for data extraction with VLMs. For Claude 3 Opus, the application programming interface (API) does not provide an option to customize the resolution. However, we had to use low-resolution images to incorporate all the images into the finite context window. For OpenAI's GPT series of models, we observed a failure to extract correct data in low-resolution mode (Figure 2).

Overall, the models GPT-4 Vision, GPT-4o, and GPT-4 Assistant achieve the highest precision on our dataset. As precision is the most relevant metric, only these models are suitable for use in extraction tasks. Taking also information completeness and cost into account, GPT-4o performs best on average, wherefore we used the GPT-4o model for further experiments.

3.2. Precision optimization using a random forest classifier

Even though a precision of up to $86\,\%$ might seem promising, this still corresponds to a large number of incorrect

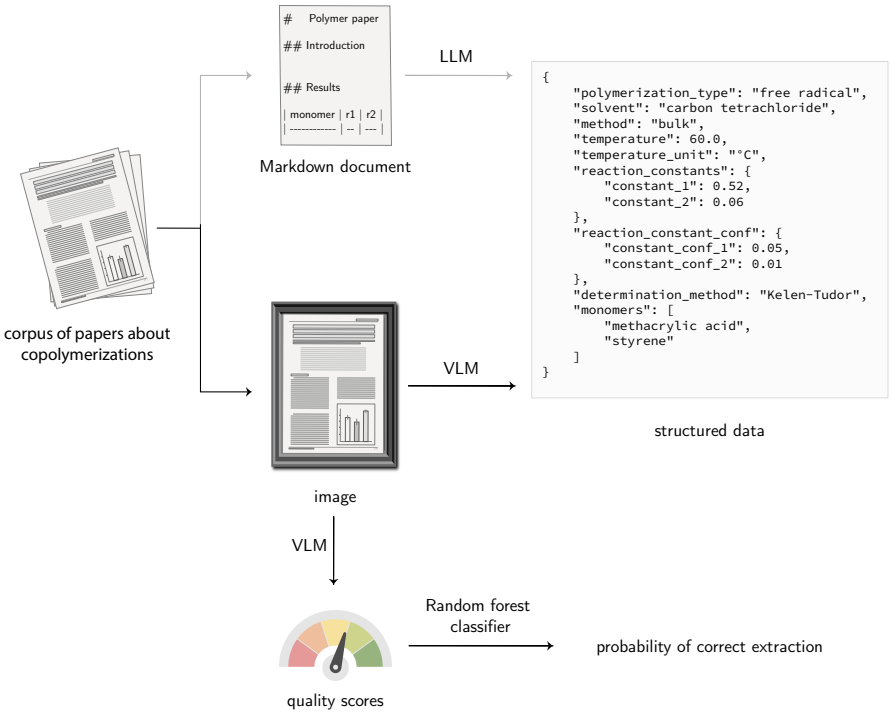

Figure 1. Overview of the data extraction workflow. In our work, we compare the extraction of data using LLMs, operating on text extracted from PDFs using VDU tools, with the performance of VLMs. We further improve the performance of the extraction workflow by developing a data-driven quality scoring approach that operates on features extracted using a VLM.

extractions when this approach is used in a high-throughput setting on thousands of papers. The error distribution on the test dataset (Figure 6) shows that the imperfect precision is mostly due to papers for which the extraction completely failed. We thus hypothesized that learning to correlate quality indicators with the expected precision can be used to identify cases where the extraction is likely to fail.

To extract quality indicators, we prompt a VLMs to return the language of the article, year of publication, number of different reactions in the article, and quality of the article in terms of readability of numbers, overall optical quality, and quality and structuredness of tables in a PDF of a paper shown as images. Afterward, we trained a random forest regressor on the quality indicators and manually extracted the precision of 35 articles. Using this model, we filtered out articles with a predicted precision below $70\,\%$ and could, in this way, increase the extraction precision from $77\,\%$ to $94\,\%$.

3.3. Analysis and visualization of the extracted polymerization data

Overall, we extracted 361 articles, including 1138 reactions, 1125 of which had r-values. We found 38 unique solvents and 218 bulk polymerizations (without any solvent).

As a correctness check of the extracted data, we additionally extracted the product of the r-values, which was provided for $16\,\%$ of the reactions. For $85\,\%$ of the reactions, the deviation between the extracted r-product and the one computed from the extracted r values is below $10\,\%$ (which is an acceptable deviation). This indicates that we can apply this as another filter to sort out reactions that likely are (partially) extracted incorrectly.

3.4. Prediction of reactivity constant products

Since the product of the reactivity constants can describe the outcome of copolymerization, we focussed on predicting it based on the monomers as well as reaction conditions. As a robust measure of generalization, we perform a split based on the monomers. This ensures that the model has not seen any of the monomers it is tested on during training. Training a model directly on the product of reactivity constants, $r_1 r_2$, is challenging because the distribution is very skewed (Figure 9). Hence, we built multiple classification models to model the relevant limiting cases which result of the different areas of the r-products (e.g. $r_1 r_2 \approx 0$: no homopolymerization of the monomers possible). For building the models, we featurized the monomers and solvents using count-based FCFPs and then built histogram-based gradient-boosting classification trees. Figure 3 shows that

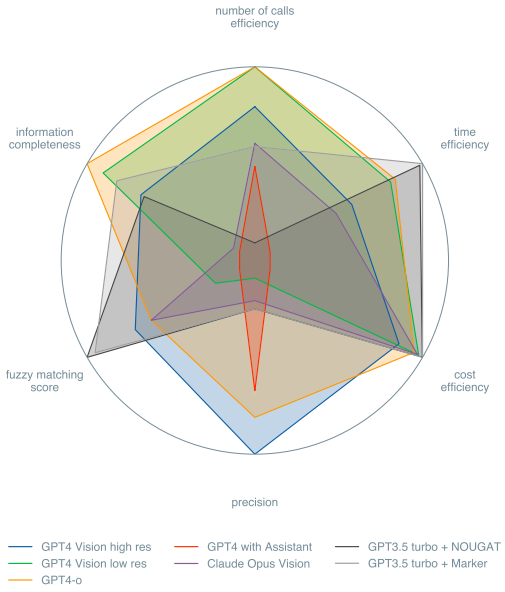

Figure 2. Plot of different important parameters of the runs of all used models; precision calculated according to Equation (1), cost efficiency calculated from the number of input and output tokens times the individual model costs, time efficiency of the model calls, number of calls efficiency for the extraction of 10 papers, information completeness calculated by dividing the number of empty entries by the total number of entries, fuzzy matching score for all entries indicating the deviation of numbers and letters; values were normalized; high values represent a desired result, e.g., a low price, low values represent undesired parameter values, e.g., a low fuzzy matching score

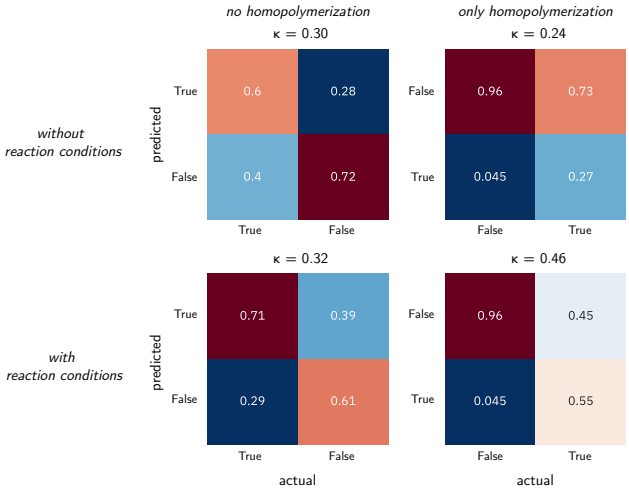

Figure 3. Confusion matrices for prediction of limiting cases of products of reactivity constants with and without information about reaction conditions. Reactant chemistry was encoded using the sum of the count-based FCFPs of the monomers. Solvents were encoded using the same fingerprints, and additionally, we added temperature as a descriptor. The top row shows the predictions of models that only include information about reactants. In the bottom row, the models also have access to the solvent and temperature, which improves predictive performance. The "no homopolymerization" case refers to $r_1 r_2 \approx 0$, whereas the case "only homopolymerization" refers to the case of $r_1 r_2 \gg 1$. As a comparison metric the Cohens Kappa κ was calculated.

our models can correctly predict the limiting copolymerization reactivity. Importantly, Figure 3 also illustrates the importance of information about the reaction conditions, without which we could only achieve limited predictive power for the limiting case of large reactivity constants that leads to homopolymerization.

3.5. Limitations

The model we have built can be further optimized in various ways. For instance, the featurization of the reactants and solvents is currently based on relatively simple fingerprints. In addition, our current models would be even more useful if they predicted the numeric values of the product of reactivity constants instead of only a class. Moreover we could improve the prediction by extracting and using an even larger copolymerization dataset. Furthermore, the extraction can likely be further improved using specialized models for table extraction in an agentic setting (Smock et al., 2021).

4. Conclusion and outlook

The ability to correctly predict the outcome of copolymerization reactions would greatly expedite chemical research. Past work has focused on modeling copolymerizations based on only reactant information. However, this has only limited predictive power. An important barrier to including reactant information has been the lack of suitable datasets, which are difficult to compile as relevant data is often reported in old, poorly digitized articles. Leveraging the GPT-4o VLM, we extracted 1138 copolymerization reactions, including reaction conditions like temperature and solvent, and used a novel quality-score-based model to improve the precision of our extraction workflow.

Our initial models have shown that information about reaction conditions markedly improves predictive performance. We envision that our further improved datasets and models will play an important role in the rational design of copolymerization reactions.

5. Methods

Article dataset We obtained the dataset of the articles from the Copolymerization Database(Takahashi et al., 2023). We downloaded PDFs of the articles using the SciDownl Python tool (Tishacy, 2023).

Evaluation pipeline For the automated comparison of different approaches, we created an automated evaluation pipeline and manually annotated 10 randomly chosen articles (33 reactions). For the scoring of the extraction, we first count the number of correctly extracted reactions. For this, we automatically convert the monomers into Simplified Molecular Input Line Entry System (SMILES) by using the chemical name resolver. (National Cancer Institute, 2023) A reaction is classified correct if the converted monomer SMILES exactly matches the one in the ground truth and the temperature, the reaction constants, and their confidence interval deviate less than $1\,\%$. We then calculated the precision of the data extraction according to Equation (1).

$$\text{precision} = \frac{\text{number of correct reactions}}{\text{total number of extracted reactions}} \quad (1)$$

Models We use GPT-3.5 Turbo in combination with the OCR-tools Nougat and Marker to obtain the articles as Markdown input. For the Nougat tool, we apply the `no-skipping` parameter to avoid false-positive conversion skips. Afterward, we directly used the obtained Markdown files without any further preprocessing. With images of the PDF file as input, we use the two vision models, Claude 3 Opus and GPT-4 Vision (with low and high-resolution modes). Additionally, we test GPT-4 Assistant (Version 1) with the PDF files as input and the "retrieval" mode. For all models, we use the same base prompt (see Listing 1) and temperature 0.

Article parsing For GPT-3.5 Turbo, we convert the articles into Markdown text and send them over to the model. Since the content window of GPT-3.5 Turbo is very limited, we chunk the input to the requested size and call the model multiple times. For GPT-4 Assistant, we pass the PDF directly to the model. For the vision models Claude 3 Opus and GPT-4 Vision, we first convert the PDF file into pictures. Afterward, we flip vertical pages in the document with the Tesseract Python library PyTesseract Contributors, 2023. At the end of each extraction call on an article, we calculated the fraction of unfilled keys in the partially extracted in our data schema. If there are more than $30\,\%$ entries empty, we call the models again to fill these up.

Random forest classifier for precision estimation We built a random forest classifier model using the scikit-learn Python package (Pedregosa et al., 2011). We optimize the hyperparameters (number of estimators, maximum depth, and minimum samples per leaf) using k-stratified cross-validation with $k = 10$ and GridSearchCV. We aim to maximize a combined accuracy and false positive rate score. We train the model with 35 manually evaluated data points.

Classification models We computed classification metrics using PyCM (Haghighi et al., 2018). For training, we randomly selected 136 monomers SMILES, resulting in 445 training points and 168 test points as we consider every reaction in which one of the training SMILES occurs as a training point.

As a proof of concept, we computed functional class counts fingerprints of the monomers and solvents using the molfeat Python package (Noutahi et al., 2023). We used a sum of the two monomer fingerprints as an input and used histogram-based gradient-boosting classification trees using `max_iter=500` and class weights of 1:50 (Pedregosa et al., 2011).

References

Ai, Q., Meng, F., Shi, J., Pelkie, B., & Coley, C. W. (2024). Extracting structured data from organic synthesis procedures using a fine-tuned large language model. *ChemRxiv preprint*. https://doi.org/10.26434/chemrxiv-2024-979fz

Blecher, L., Cucurull, G., Scialom, T., & Stojnic, R. (2023). Nougat: Neural optical understanding for academic documents.

Center for Research on Foundation Models. (2024). The first steps to holistic evaluation of vision-language models [Accessed: 2024-06-14].

Coley, C. W., Jin, W., Rogers, L., Jamison, T. F., Jaakkola, T. S., Green, W. H., Barzilay, R., & Jensen, K. F. (2019). A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, *10*(2), 370–377.

Cowie, J., & Arrighi, V. (2007, July). *Polymers* (3rd ed.). CRC Press.

Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., & Jain, A. (2024). Structured information extraction from scientific text with large language models. *Nature Communications*, *15*(1). https://doi.org/10.1038/s41467-024-45563-x

Dossi, M., & Moscatelli, D. (2012). A qm approach to the calculation of reactivity ratios in free-radical copolymerization. *Macromolecular Reaction Engineering*, *6*(2–3), 74–84. https://doi.org/10.1002/mren.201100065

Farajzadehahary, K., Telleria-Allika, X., Asua, J. M., & Ballard, N. (2023). An artificial neural network to predict reactivity ratios in radical copolymerization. *Polymer Chemistry*, *14*(23), 2779–2787.

Fazakas-Anca, I. S., Modrea, A., & Vlase, S. (2021). Determination of reactivity ratios from binary copolymerization using the k-nearest neighbor nonparametric regression. *Polymers*, *13*(21), 3811.

Filley, J., McKinnon, J. T., Wu, D. T., & Ko, G. H. (2002). Theoretical study of ethylene-vinyl acetate free-radical copolymerization: Reactivity ratios, penultimate effects, and relative rates of chain transfer to polymer. *Macromolecules*, *35*(9), 3731–3738. https://doi.org/10.1021/ma011805+

Gao, H., Struble, T. J., Coley, C. W., Wang, Y., Green, W. H., & Jensen, K. F. (2018). Using machine learning to predict suitable conditions for organic reactions. *ACS central science*, *4*(11), 1465–1476.

Haghighi, S., Jasemi, M., Hessabi, S., & Zolanvari, A. (2018). Pycm: Multiclass confusion matrix library in python. *Journal of Open Source Software*, *3*(25), 729. https://doi.org/10.21105/joss.00729

Hawizy, L., Jessop, D. M., Adams, N., & Murray-Rust, P. (2011). Chemicaltagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics*, *3*(1). https://doi.org/10.1186/1758-2946-3-17

Jablonka, K. M., Ai, Q., Al-Feghali, A., Badhwar, S., Bocarsly, J. D., Bran, A. M., Bringuier, S., Brinson, L. C., Choudhary, K., Circi, D., Cox, S., de Jong, W. A., Evans, M. L., Gastellu, N., Genzling, J., Gil, M. V., Gupta, A. K., Hong, Z., Imran, A., ... Blaiszik, B. (2023). 14 examples of how llms can transform materials science and chemistry: A reflection on a large language model hackathon. *Digital Discovery*, *2*(5), 1233–1250. https://doi.org/10.1039/d3dd00113j

Jorner, K., Tomberg, A., Bauer, C., Sköld, C., & Norrby, P.-O. (2021). Organic reactivity from mechanism to machine learning. *Nature Reviews Chemistry*, *5*(4), 240–255. https://doi.org/10.1038/s41570-021-00260-x

Lewis, F. M., Walling, C., Cummings, W., Briggs, E. R., & Mayo, F. R. (1948). Copolymerization. iv. effects of temperature and solvents on monomer reactivity ratios. *Journal of the American Chemical Society*, *70*(4), 1519–1523.

Moosavi, S. M., Chidambaram, A., Talirz, L., Haranczyk, M., Stylianou, K. C., & Smit, B. (2019). Capturing chemical intuition in synthesis of metal-organic frameworks. *Nature Communications*, *10*(1). https://doi.org/10.1038/s41467-019-08483-9

National Cancer Institute. (2023). Chemical identifier resolver documentation.

Nguyen, T., & Bavarian, M. (2023). Machine learning approach to polymer reaction engineering: Determining monomers reactivity ratios. *Polymer*, *275*, 125866.

Noutahi, E., Wognum, C., Mary, H., Hounwanou, H., Kovary, K. M., Gilmour, D., Thibaultvarin-R, Burns, J., St-Laurent, J., T, DomInvivo, Saurav Maheshkar, & Rbyrne-Momatx. (2023). Datamol-io/molfeat: 0.9.4. https://doi.org/10.5281/ZENODO.8373019

Paruchuri, V. (2023). Marker: Open source machine learning model for data annotation.

Patiny, L., & Godin, G. (2023). Automatic extraction of fair data from publications using llm. *ChemRxiv preprint*. https://doi.org/10.26434/chemrxiv-2023-05v1b-v2

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Polak, M. P., & Morgan, D. (2024). Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, *15*(1). https://doi.org/10.1038/s41467-024-45914-8

PyTesseract Contributors. (2023). Pytesseract: Python-tesseract is an optical character recognition (ocr) tool for python. that is, it will recognize and "read" the text embedded in images.

Raccuglia, P., Elbert, K. C., Adler, P. D. F., Falk, C., Wenny, M. B., Mollo, A., Zeller, M., Friedler, S. A., Schrier, J., & Norquist, A. J. (2016). Machine-learning-assisted materials discovery using failed

experiments. *Nature*, *533*(7601), 73–76. https://doi.org/10.1038/nature17439

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C., & Lee, A. A. (2019). Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, *5*(9), 1572–1583. https://doi.org/10.1021/acscentsci.9b00576

Segler, M. H., Preuss, M., & Waller, M. P. (2018). Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, *555*(7698), 604–610.

Shields, B. J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J. I. M., Janey, J. M., Adams, R. P., & Doyle, A. G. (2021). Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, *590*(7844), 89–96. https://doi.org/10.1038/s41586-021-03213-y

Smock, B., Pesala, R., & Abraham, R. (2021). Pubtables-1m: Towards comprehensive table extraction from unstructured documents.

Swain, M. C., & Cole, J. M. (2016). Chemdataextractor: A toolkit for automated extraction of chemical information from the scientific literature. *Journal of Chemical Information and Modeling*, *56*(10), 1894–1904. https://doi.org/10.1021/acs.jcim.6b00207

Takahashi, K.-i., Mamitsuka, H., Tosaka, M., Zhu, N., & Yamago, S. (2023). Copoldb: A copolymerization database for radical polymerization. https://doi.org/10.26434/chemrxiv-2023-t7v1b

Tishacy. (2023). Scidownl github repository.

Vorob'eva, A., Ablyakimov, E., Leplyanin, G., Rafikov, S., & Gladyshev, G. (1974). Study of the kinetics of polymerization and copolymerization of methacrylic acid sulpholanate. *Polymer Science U.S.S.R.*, *16*(2), 405–411. https://doi.org/10.1016/0032-3950(74)90567-x

6. Acknowledgement and Disclosure of Funding

The research was supported by the Carl-Zeiss Foundation as well as Intel and Merck via the AWASES research center and the "Talent Fund" of the LIFE profile line of the Friedrich-Schiller University of Jena. We thank Adrian Mirza for preparing the original article dataset. Additionally, we thank Santiago Miret and Vijay Narasimhan for useful discussions.

A. Appendix / supplemental material

A.1. Paper corpus description

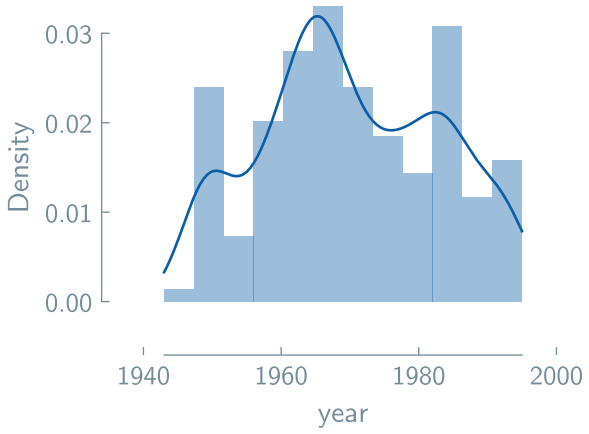

Figure 4. Distribution of publication year of papers in our corpus.

Figure 4 shows the distribution of publication years. The articles in the database were published between 1940 and 1990. Since all articles in the database were published before 2000, publishers only provide PDF documents for download, whereas today's practice has changed, and publishers also provide machine-readable formats like HTML files, which simplifies processing and data mining.

Additionally, the required information can be found in various sections of the articles. Some articles provide all necessary data in an easily accessible form in the abstract, while in other articles, the data is spread through several sections, tables, and figures. Further, the resolution and quality of the provided PDF files are mostly poor, and tables are especially difficult to read (Figure 5).

TABLE 3. THE MAS (M_1) COPOLYMERIZATION WITH SOME VINYL MONOMERS (M_2)
([AIBN] = $6{\cdot}1 \times 10^{-4}$ mole/l.)

M_2	M_2, mole%		r_1	r_2	$r_1 \cdot r_2$	Q	e	φ
	at start	in copolymer						
Styrene	97·14	88·87	1 5±0·1	0·22±0 01	0 33	2·0	0·26	25
	94 06	80·16						
	75·8	63·8						
MM	96 21	96·14	3 34±0 25	0 67±0 02	2 24	2 5	0 4	5
	92 06	87 20						
	80 9	68 60						
MA	96 6	95 9	1 0±0 1	0·83±0 02	0 83	—	—	600
	89 5	87 9						
	84 7	82 7						

Figure 5. Example of a representative table found in the PDF dataset (Vorob'eva et al., 1974).

A.2. Details on data extraction workflow

We use the same base prompt for all the used models Listing 1. We used a temperature of 0 for all model calls.

Listing 1. Base prompt used for the data extraction

```
The content of the Markdown is a scientific paper about copolymerization of
    monomers .
```

```
We only consider copolymerizations with 2 different monomers. If you find a
    polymerization with just one or more than 2 monomers ignore them.
Its possible, that there is also the beginning of a new paper about polymers in
    the PDF.
Ignore these. In each paper there could be multiple different reaction with
    different pairs of monomers and same reactions with different reaction
    conditions.
The reaction constants for the copolymerization with the monomer pair is the most
     important information. Be careful with numbers and do not miss the decimal
    points.
If there are polymerization's without these constants, ignore these.
From the PDF, extract the polymerization information from each polymerization and
     report it in valid json format.
Also pay attention to the caption of figures.
Don't use any abbreviations, always use the whole word.
Try to keep the string short. Exclude comments out of the json output. Return one
     json object.
Stick to the given output datatype (string, or float).

Extract the following information:

reactions: [
    {
        "monomers": ["Monomer 1", "Monomer 2"] as STRING (only the whole Monomer
            name without abbreviation)
        "reaction_conditions": [
            {
                "polymerization_type": polymerization reaction type (free radical
                    , anionic, cationic, ...) as STRING,
                "solvent": used solvent for the polymerization reaction as STRING
                     (whole name without
                         abbreviation, just name no further details like 'sulfur
                             or water free'); if the solvent is water put just "
                             water"; ,
                "method": used polymerization method (solvent(polymerization
                    takes place in a solvent), bulk (polymerization takes place
                    without any solvent, only reactants like monomers built the
                    reaction mixture), emulsion...) as STRING,
                "temperature": used polymerization temperature as FLOAT ,
                "temperature_unit": unit of temperature (°C, °F, ...) as STRING,
                "reaction_constants": { polymerization reaction constants r1 and
                    r2 as FLOAT (be careful and just take the individual values,
                    not the product of these two),
                "constant_1":
                "constant_2": },
                "reaction_constant_conf": { confidence interval of polymerization
                     reaction constant r1 and r2 as FLOAT
                "constant_conf_1":
                "constant_conf_2": },
                "determination_method": method for determination of the r-values
                    (Kelen-Tudor, EVM Program...) as STRING
            },
            {
                "polymerization_type":
```

```
                "solvent":
                ...
            }
        ]
    },
    {
        "monomers":
            "reaction_condition": [
                { ... }
            ]
    }
    "source": doi url or source as STRING (just one source)
    "PDF_name": name of the pdf document
]


If the information is not provided put null.
If there are multiple polymerization's with different parameters report as a
    separate reaction (for different pairs of monomers) and reaction_conditions (
    for different reaction conditions of the same monomers).
```

Detailed metrics are plotted in Figure 2 and listed in Table 1.

Table 1. Comparison of models with standard deviation. Metrics were obtained as an average over three model runs with the 10 example articles; number of model calls and execution time in s are summed up for the 10 articles; cost is calculated by multiplying the sum of input and output token with the individual costs per token; precision, fuzzy matching score and rate of empty entries are the average over the 10 articles. For all models, we used a temperature of 0.

Model	Number of Calls	Execution Time	Cost	Precision	Fuzzy Matching Score	Rate of Empty Entries
GPT-4 Vision high res	14.00 ± 1.73	537.81 ± 44.76	1.51 ± 0.06	0.81 ± 0.10	0.73 ± 0.01	13.99 ± 1.64
GPT-4 Vision low res	10.00 ± 0.00	328.00 ± 25.78	0.29 ± 0.01	0.00 ± 0.00	0.63 ± 0.01	9.53 ± 1.76
GPT-4 Assistant	20.00 ± 2.00	978.97 ± 330.56	9.50 ± 1.64	0.71 ± 0.09	0.60 ± 0.02	25.62 ± 3.93
Claude 3 Opus	17.67 ± 0.58	622.13 ± 19.86	0.37 ± 0.00	0.10 ± 0.04	0.71 ± 0.01	24.90 ± 2.67
GPT-3.5 Turbo + Nougat	27.67 ± 1.53	170.67 ± 5.28	0.07 ± 0.00	0.19 ± 0.03	0.79 ± 0.01	14.36 ± 1.19
GPT-3.5 Turbo + Marker	18.00 ± 0.00	115.73 ± 14.10	0.05 ± 0.00	0.20 ± 0.00	0.78 ± 0.00	11.14 ± 1.04
GPT-4o	10.00 ± 0.00	302.85 ± 15.47	0.58 ± 0.00	0.68 ± 0.03	0.71 ± 0.00	7.64 ± 0.21

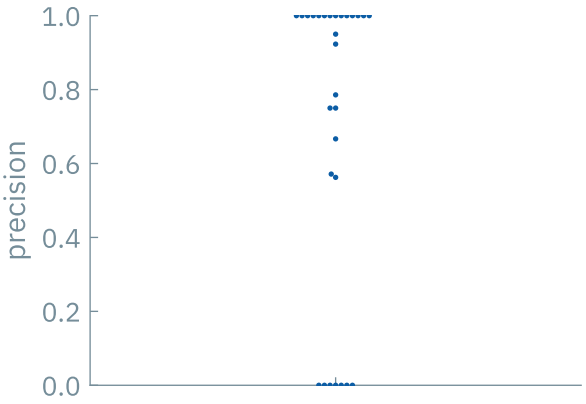

Figure 6. Distribution of precision in the test dataset prior to filtering using the random-forest regressor.

Table 2. Precision of the data extraction of the different data test sets.

Dataset	Number of Papers	Number of Reactions	Precision
Test Dataset	42	255	77 %
Filtered Test Dataset	34	199	94 %

A.3. Overview of extracted data

For the 1138 unique extracted reactions, we observe 38 unique solvents depicted in Figure 7.

We extracted 47 unique polymerization temperatures shown in Figure 8.

We calculate the r-products of the copolymerization reaction by multiplying the extracted r-vales. The calculated products range from -13 to 268. The distribution of the r-products is shown in Figure 9. The distribution is very skewed with long tails.

In the extracted dataset (787 reactions with non-null monomers, solvent, temperature, and r-values), certain reactions (same monomer, solvent, and temperature) occur multiple times. 69 reactions are duplicated, 11 occur three times, 2 four times, and one reaction occurs five times. From those, we can obtain a measure of the variance in the data (Figure 10). In only $4.4\,\%$ of the cases, the standard deviation (σ) is larger than half the value of the mean (μ). In $5.23\,\%$ of the cases, the standard deviation is larger than one-tenth of the mean.

For the training of the machine learning model, we removed those entries (leaving us with 623 entries). For machine learning, we used the mean-aggregated values.

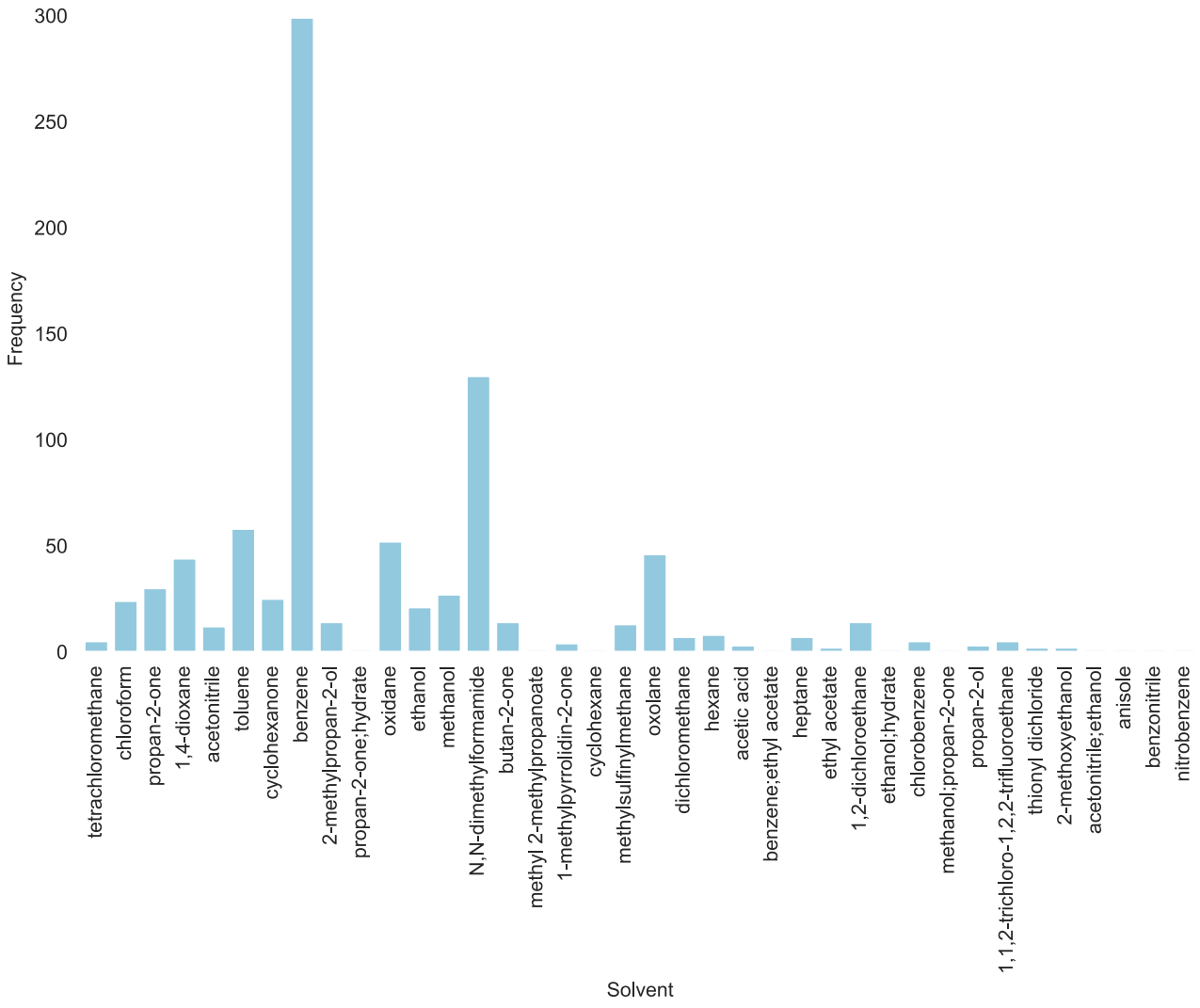

Figure 7. Distribution of the solvents used in the extracted copolymerization reactions.

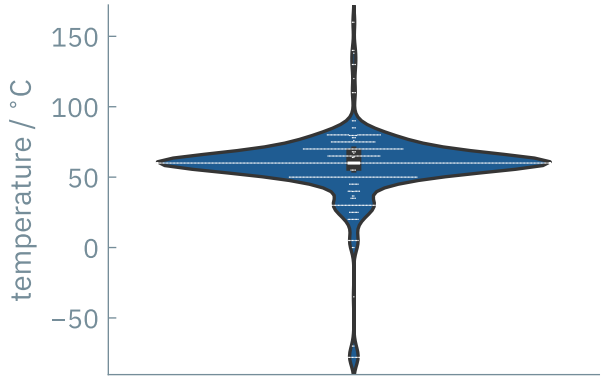

Figure 8. Distribution of the temperatures in °C used in the extracted copolymerization reactions.

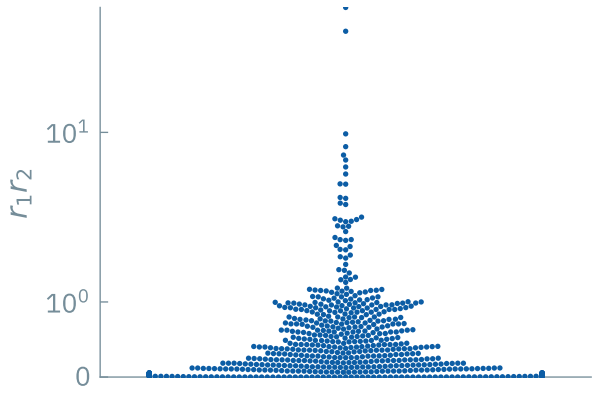

Figure 9. Distribution of the r-products calculated by multiplying the extracted r-values.

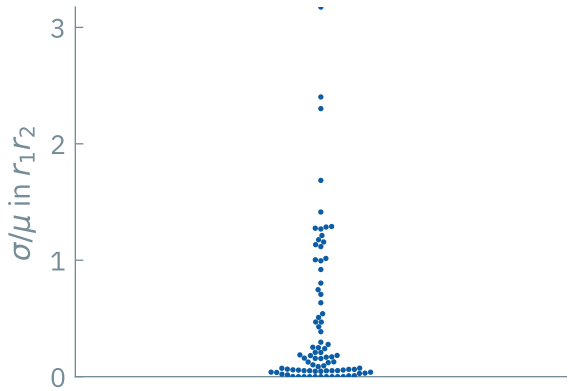

Figure 10. Distribution of normalized standard deviation for repeated reactions in our extracted dataset.