UNIVERSITY OF OXFORD

Oxford Economic and Social History Working Papers

Number 228, March 2026

Multimodal LLMs for Historical Dataset Construction from Archival Image Scans: German Patents (1877-1918)

NICLAS GRIESSHABER & JOCHEN STREB

MULTIMODAL LLMS FOR HISTORICAL DATASET CONSTRUCTION FROM ARCHIVAL IMAGE SCANS: GERMAN PATENTS (1877–1918)*†‡

Niclas Griesshaber§
University of Oxford
niclas.griesshaber@history.ox.ac.uk

Jochen Streb
University of Mannheim
jochen.streb@uni-mannheim.de

December 22, 2025

ABSTRACT

We leverage multimodal large language models (LLMs) to construct a dataset of 306,070 German patents (1877–1918) from 9,562 archival image scans using our LLM-based pipeline powered by Gemini-2.5-Pro and Gemini-2.5-Flash-Lite. Our benchmarking exercise provides tentative evidence that multimodal LLMs can create higher quality datasets than our research assistants, while also being more than 795 times faster and 205 times cheaper in constructing the patent dataset from our image corpus. About 20 to 50 patent entries are embedded on each page, arranged in a double-column format and printed in Gothic and Roman fonts. The font and layout complexity of our primary source material suggests to us that multimodal LLMs are a paradigm shift in how datasets are constructed in economic history. We open-source our benchmarking and patent datasets as well as our LLM-based data pipeline, which can be easily adapted to other image corpora using LLM-assisted coding tools, lowering the barriers for less technical researchers. Finally, we explain the economics of deploying LLMs for historical dataset construction and conclude by speculating on the potential implications for the field of economic history.

Keywords Multimodal Large Language Models · Information Extraction · Dataset Construction · German Patents

1 Introduction

For a long time, economic historians focused on compiling and interpreting long macroeconomic time series, with GDP per capita playing the most prominent role (Maddison, 2006; Broadberry et al., 2015). This inevitably led to methodological limitations, as microeconomic data is needed to understand how consumers and producers react to economic policy changes or exogenous shocks. In particular, it is important to consider the heterogeneity of historical actors. People differ in terms of age, gender, education, social status, income, and wealth, among other things, and depending on these differences, they make different decisions under the same external conditions. In order to take this heterogeneity into account, large-scale microeconomic datasets are required.

The fact that economic historians rarely use microeconomic data are not due to their inaccessibility. Socioeconomic data on consumers can be obtained, for example, from historical census data (Long & Ferrie, 2013; Ruggles, 2014; Abramitzky et al., 2021), savings books (Lehmann-Hasemeyer & Streb, 2018), information on companies from patent

*Paper website: https://historymind.ai
†Code: https://github.com/niclasgriesshaber/llm_patent_pipeline.git
‡Archival Image Corpus: https://digi.bib.uni-mannheim.de/sammlungen/patentregister
§Corresponding author.

statistics (Donges & Streb, 2024), stock market newspapers (Lehmann-Hasemeyer & Opitz, 2024), city directories (Albers & Kappner, 2023), or trade registers (Guinnane et al., 2007). However, accessing these data was associated with high and sometimes prohibitive costs because the manual construction of the dataset by research assistants was slow, expensive, and error-prone. To accelerate historical dataset construction, methods from the digital humanities and computer science, such as optical character recognition (OCR), named-entity recognition (NER), or information extraction (IE), have become increasingly attractive within the field of economic history (Shen et al., 2020; Dell et al., 2023; Bergeaud & Verluise, 2024). However, these digital methods require advanced programming and labeled data to build custom pipelines that work only on homogeneous image corpora. Recent advances in artificial intelligence (AI) have the potential to remove these barriers and make large-scale dataset construction accessible to non-technical researchers, thereby eliminating the field's major bottleneck: transforming archival image scans into a structured dataset.

Multimodal large language models (LLMs) are neural networks that can jointly process multiple types of data. For example, in this paper we send images and raw text instructions to multimodal LLMs to construct a large-scale historical patent dataset. Multimodal LLMs have shown remarkable performance across a wide range of tasks, including language and vision (Devlin et al., 2019; Brown et al., 2020; Dosovitskiy, 2020). They are—like many of the existing methods from OCR, NER, and IE—based on the deep learning paradigm, which refers to neural networks with many layers that can learn complex patterns from data (LeCun et al., 2015; Goodfellow et al., 2016). When LLMs are scaled with data, parameters, and computing capacity, they become surprisingly powerful (Kaplan et al., 2020; Hoffmann et al., 2022; Wei et al., 2022), offering a unified approach to OCR, NER, and IE. These immense economies of scale are also the reason why the most capable LLMs are almost exclusively developed by private AI companies.

The advancements and commercialization of multimodal large language models have enabled us to make several contributions. First, to construct a dataset that contains all 306,070 patent entries that are included in 41 physical volumes of historical patent registers published between 1877 and 1918, we used LLM-assisted coding tools to develop an LLM-based data pipeline that processed our 9,562 image scans. We then extracted five variables from each patent entry: *patent_id*, *assignee*, *location*, *title*, and *date*. Moreover, our research assistants manually created a *student-constructed* benchmarking dataset from 41 randomly selected images, one from each of the annual patent registers used. Initially, we compared the *LLM-generated* dataset against the *student-constructed* dataset to evaluate the reliability of our LLM-based pipeline—especially to check for hallucinations that would undermine the data quality and bias any subsequent econometric analysis. Surprisingly, we realized that many highlighted differences between both datasets were not due to errors made by the LLM, but due to mistakes made by our research assistants. For this reason, we also constructed a second, *perfect* benchmarking dataset. By comparing the *student-constructed* and *LLM-generated* dataset to the *perfect* dataset, we provide tentative evidence that multimodal LLMs are able to produce higher quality datasets from our image corpus than our research assistants. Finally, we open-source both our LLM-based data pipeline and the new historical patent dataset to accelerate empirical research in economic history.

This paper is structured as follows: Section 2 provides the historical background of our data. Section 3 then reviews related work on the construction of patent databases, including manual work, custom machine learning approaches, and the emerging literature on LLMs in economic history. Section 4 describes our primary sources that form our archival image corpus. Section 5 presents our LLM-based data pipeline to construct the patent dataset, followed by Section 6 which addresses data quality and hallucination concerns using our benchmarking datasets. Section 7 explains the economics of constructing datasets from image scans using multimodal LLMs. Section 8 concludes and discusses the potential implications of multimodal LLMs for the field of economic history.

2 Historical Background

Modern economic growth, which began in Great Britain in the early nineteenth century and has since been emulated by many countries around the world, is based on a never-ending stream of innovation. A central question in economic history is therefore why it took so long for innovations to evolve from a rare occurrence to a constant driver of growth (Galor, 2011; Mokyr, 2011). To understand which factors promote and hinder the emergence of innovation, it is first necessary to measure the type and scope of innovations as precisely as possible. In economic history research, this is done primarily on the basis of patent statistics.

The origins of the patent system date back at least to mid-fifteenth century Venice. However, it was not until the end of the eighteenth century that patents evolved from a privilege into an intellectual property right that could be obtained by registration with a patent authority. All patent authorities charged a filing fee for granting a patent; some also examined the novelty of a registered invention as an additional requirement. Since then, a patent holder has had the exclusive right to manufacture and market the protected innovation within the geographical scope of the patent law and for a limited period of time. For patent protection to work, potential imitators must be informed about which innovations are protected. Patent authorities have therefore long provided information about patents granted and have also collected detailed patent descriptions so that interested parties can consult them if necessary (Moser, 2011; Cox, 2019). These historical sources are still available today and allow databases to be created with microeconomic information about the inventor and co-inventors, their gender and place of residence, as well as the number, title, and technological class of the patent.

The comparative ease of access to this mass of data is the great advantage of patents as a measure of the extent and direction of technological progress. Critics of this approach like to point out that not all innovations are patented, either because patent law does not allow them to be or because inventors believe that innovation gains are more likely to be realized through secrecy. Added to this is the problem that pure patent counts neglect the fact that only a minority of patents granted protect valuable innovations, while most patents represent rather worthless ideas that hardly constitute incremental innovations (Griliches, 1990; Moser, 2012). The first criticism concerns an undeniable weakness of patent statistics as a measure of innovation, which has not yet been remedied but is accepted by most economic historians. The alternative method proposed by Petra Moser, which is to identify historical innovations with the help of world exhibition data, has its own shortcomings (Moser, 2005; Domini, 2020). The second criticism has since lost importance because economic historians have found various ways to assess the quality of patents (Streb, 2024).

3 Related Work

3.1 Manual Dataset Construction

Constructing datasets from primary sources by hand is very labor-intensive. Over the past decades, a large number of historical patent databases have been created manually. These databases now exist for the early industrialized countries of Great Britain, France, the United States, and the Netherlands, for the Mediterranean countries of Italy and Spain, for the Scandinavian countries of Finland, Norway, and Sweden, for Japan and even for some Latin American countries such as Argentina, Cuba, and Mexico (Streb, 2023).

In the mid-2000s, Streb et al. (2006) decided to create a historical patent database for the German Empire and the Weimar Republic. However, they did not have sufficient financial resources to pay all the research assistants who would have been needed to manually transfer the more than half a million patents granted in Germany between 1877 and 1932 into a digital database. As a result, the economic historians had to forgo completeness and restricted their database to a subset that only included the most valuable patents. Because the German patent system levied an annual patent renewal fee, which was intended to encourage patent holders to abandon unprofitable patents as quickly as possible rather than retaining them for the maximum term of 15 years, it is possible to infer the economic value of a patent from its life span. They selected patents for the database that had been held for at least ten years from the date of grant. The resulting database comprises around 68,700 long-lived patents, which corresponds to about 10% of all patents granted in the German Empire and the Weimar Republic. Despite the considerable reduction in the amount of data, it took several months to manually extract all the relevant information, perfectly illustrating the scalability limits of manual dataset construction.

3.2 Custom OCR and NER Pipelines

To overcome the constraints of manual dataset construction, recent work has utilized narrow task-specific models from machine learning. With regard to the creation of patent databases, “PatentCity” represents the state-of-the-art approach in this paradigm. Bergeaud & Verluise (2024) use traditional OCR algorithms and train their own deep learning models

to construct a patent database. Due to this transition from manual to digital methods, the amount of data they were able to collect exceeds all previous ones. Their database contains information on the assignees and inventors of all patents granted in Germany between 1877 and 1980, in France between 1903 and 1980, in the United Kingdom between 1893 and 1980, and in the United States between 1836 and 1980. Where specified, the inventor's profession, citizenship, and address were also recorded. Furthermore, they assigned the inventors' addresses to a county or a municipality.

To construct the PatentCity database, the authors first converted the historical patent specifications available as image scans into editable text files using Tesseract v5.0 and their own in-house OCR algorithm. In a second step, they trained a custom NER model to extract the relevant variables from the OCR text and applied an relationship algorithm to reconstruct how the entities interrelate. To train the NER model, the authors had to create a manually annotated dataset that was split into a training and test set. The former was used to teach the deep-learning model how to extract the desired entities. After the training had finished, they evaluated the model's performance by comparing its predictions with the ground truth on the test set.

However, such narrow machine learning approaches face their own bottlenecks. The method by Bergeaud & Verluise (2024) is not universal and they had to retrain their NER model whenever the layout of the image scans changed because the structure of the OCR text shifted accordingly. This always required a new annotated dataset. Moreover, much of the spatial relationship of the embedded text on an image is lost when they perform OCR, which is crucial when constructing datasets from more complex historical documents depicting double-column layouts, tables, or maps. Finally, LLM-assisted coding tools did not exist when they first published their work, making it very difficult for non-technical researchers to adopt their pipeline to other image corpora.

The arrival of multimodal LLMs may offer a universal solution for historical documents as it mitigates all of the aforementioned issues, which enabled us to partially replicate the PatentCity database for German patents from 1877 to 1918 despite using a different primary source. Instead of patent specifications depicting information on a single patent per image scan, our source contains between 20 to 50 short patent entries per page, which are arranged in a double-column format. Furthermore, we go beyond replication by providing all patent titles that are not included in the PatentCity database.

3.3 Large Language Models in Economic History

There is an emerging literature on large language models for text-as-data approaches in economics and economic history (e.g., Bartik et al., 2025; Chyn et al., 2025; Griesshaber & Ogilvie, 2025; Lagakos et al., 2025; Sockin et al., 2025). Existing work uses LLMs to extract higher-level features from editable text to construct variables for downstream econometric analysis. For example, Griesshaber & Ogilvie (2025) use GPT-4o to classify sentences into broader economic categories to examine institutional differences within the guild system across colonial Latin America. Lagakos et al. (2025) extract features from biographies to understand the sources of a meaningful life in early-twentieth-century America. In both cases, LLMs lead to unprecedented productivity increases as economic historians and research assistants do not have to read and manually classify the thousands of biographies and guild regulations. However, as LLMs were trained on a finite dataset, they inevitably make biased predictions, especially when those predictions involve value judgments. Carlson & Dell (2025) show that this bias can propagate to downstream estimators. They further emphasize that researchers may use various LLMs with different biases to achieve significant results in their regression analyses. For this reason, the authors propose the MAR-S framework, which includes the construction of a validation dataset to correct for the bias introduced during the LLM's feature extraction.

Compared to these text-as-data approaches, value judgments are usually less pronounced when merely transcribing printed, Gothic or handwritten text from image scans. Multimodal LLMs can be given an image together with a text instruction, prompting them to transcribe the embedded text. There are several factors that influence the LLMs' transcription accuracy: font type, image size, information density, resolution, document degradation, prompt, and many more. As the immense heterogeneity of archival sources affects all of these factors, it is difficult to assess the general reliability of multimodal LLMs on text transcription from image scans. Greif et al. (2025) benchmark the OCR capabilities of GPT-4o, Gemini-2.0-Flash, and Transkribus' Text Titan I on eighteenth- and nineteenth-century German city directories, printed in Roman and Gothic. They report the best transcription results for Gemini-2.0-Flash. Humphries et al. (2025) find that LLMs transcribe eighteenth- and nineteenth-century English handwritten documents

with higher accuracy than fine-tuned models by Transkribus. Levchenko (2025) also reports this for Russian prints in Civil font, with Gemini-2.5-Pro yielding the best performance among 12 evaluated multimodal LLMs. Crosilla et al. (2025) show the superiority of LLMs over Transkribus on modern handwriting. Even more astonishing than the universal transcription capabilities of multimodal LLMs is the fact that we still do not fully understand how they accomplish this task. As a result, the nascent subfield of mechanistic interpretability tries to cast light on the internal mechanisms behind this ability (Baek et al., 2025; Kim et al., 2025).

Perhaps even more striking than the ability of multimodal LLMs to transcribe embedded text in images is that they can be prompted to extract only the desired information in order to construct a structured dataset. Intuitively, strong transcription capabilities are a prerequisite for this task. The economic historian Jonathan Jayes (2025) employs Gemini-2.0-Flash to extract information from image scans of Swedish firm reports. As more and more economic historians use multimodal LLMs to create large-scale datasets, it will be necessary to benchmark multimodal LLMs rigorously on the task of information extraction, and by extension, on the task of historical dataset construction. Xie et al. (2025, p. 1) find that GPT-4o "generates usable yet imperfect data" when deployed to extract information from Swedish patent cards (1945–1975). Luo (2025) expands the literature on information extraction on Swedish patent cards by benchmarking several open-source and proprietary LLMs on the same sample, with Gemini-2.5-Pro achieving the strongest performance. Likewise, parallel work by Vafaie et al. (2025) evaluates GPT-4o-mini and open-source models like InternVL2.5 for key information extraction on a new annotated dataset from heterogeneous German index cards. However, the accuracy of all these models is insufficient for robust research in economic history. Moreover, a report on work in progress by Rodrigues et al. (2025) finds that Gemini-2.5-Flash outperforms GPT-4o when creating JSON entries for two-column printed bibliographies. Finally, Bäcker-Peral et al. (2025) developed an LLM-based pipeline to extract information from historical tables to build a panel dataset. They also manually construct a dataset and show that their downstream regression analyses are "statistically indistinguishable whether using LLM or gold standard data" (Bäcker-Peral et al., 2025, p. 1). They claim that multimodal LLMs offer a "watershed change for the digitization of historical tables" (Bäcker-Peral et al., 2025, p. 1), which is even more striking as they use Claude-3.5-Sonnet and Gemini-1.5-Pro—multimodal LLMs that are much weaker than the models we use in this paper.[1]

4 Data

In Germany, a nationwide patent law was not introduced until 1877, replacing the numerous independent patent laws of the German states that had previously been in force (Donges & Selgert, 2019). To provide the interested public with an overview of the newly granted patents, the Reich Patent Office published an annual volume listing all patents entered into the register in the previous calendar year. This annual patent register did not list the newly granted patents in numerical order. The order of the patents was based on the 89 different technological classes used by the Patent Office. First, all patents in class 1, "Preparation of ores", were listed in numerical order, followed by all patents in class 2, "Baking" (for an example, see page 312 in Donges & Streb, 2024), and the list always ended with patents in technological class 89, "Sugar production". Figure 1 depicts a representative page of our corpus. Several pieces of information were provided for each of these patents: the patent number, the name and address of the patent holder and, in the case of foreign patent holders, the name and address of their legal representative in Germany, the title of the patent, and the date of application, to which a priority claim is appended in rare cases (Figure 2).

We were able to track down all volumes that were issued by the Reich Patent Office from 1877 until 1918. In total, we collected 41 physical books that were scanned by the Research Data Center at the University of Mannheim at a resolution of 300 dpi per image. A completeness check was conducted to ensure that not a single page was inadvertently skipped after which the TIFF images were merged into a PDF. For our dataset, we were only interested in each volume's chapter "Systematische Übersicht" that provides the list of newly granted patents described above. The layout of this chapter is fairly consistent across all 41 volumes. Patent entries are always short paragraphs, which are arranged in this double-column format. Likewise, the beginning of each of the 89 technological classes is introduced as a heading within one of the two main content columns. Finally, every page has a running header with the current patent number and technological class.

[1] Based on this work, fellow researchers at the Federal Reserve Bank of Philadelphia have released a contemporaneous report on the institutional potential of multimodal LLMs for historical data collection (Moulton & Severen, 2025).

FIGURE 1
THE PRIMARY SOURCE

gerippter oder gewellter Unterfläche; Zus. z. Pat. 92599. 8/3 98. — A 733.

99818. Weiß, H., Hannover, Friesenstr. 22. Feuerzeug mit Windschutz und Cigarrenabschneider. 10/3 98. — A 879.

99891. Schaefer, H., Werdohl. Von zwei Seiten zu tragende Brosche. 26/3 98. — A 898.

100580. Lohmann & Welschehold, Meinerzhagen. Verschlußvorrichtung für Gürtelschnallen u. dgl. 25/3 98.

100738. George, A., Berlin S.O., Köpenickerstr. 47. Für den gleichmäßigen Vorschub gelochter Zündstreifen; Zus. z. Pat. 96186. 20/4 98.

100739. Kiehnle, A., Pforzheim, Theaterstr. 14. Elastisches Armband. 1/5 98.

100998. Berg, W., Lüdenscheid. Hosenträger-Klemmschnalle. 4/11 97.

101042. Mandelbaum & Kohn, Wien; Vertr.: C. F. Reichelt, Berlin N.W., Luisenstr. 26. Stulpenknopf mit doppelseitigem Klappverschluß. 30/4 97.

101160. Proskauer, J., Leipzig, Färberstr. 12. Maschine zum Aufheften von Zwei- und Vierlochknöpfen mittels Draht. 9/12 97.

Klasse 45. Land- und Forstwirthschaft, Garten- und Weinbau, Zootechnik.

Vgl. die Klasse 6, 16, 17, 30, 34, 37, 50, 53, 56, 59, 63, 82, 89.

94926 A 33. — 94930 A 98. — 95035 A 13. — 95171 A 130. — 95172 A 80. — 95173 A 47. — 95263 A 47. — 95277 A 48. — 95327 A 61. — 95328 A 48. — 95329 A 62. — 95369 A 48. — 95475 A 168. — 95577 A 150. — 95594 A 130. — 95656 A 130. — 95800 A 112. — 95847 A 168. — 95848 A 192. — 95849 A 215. — 95850 A 112. — 95959 A 192. — 95968 A 215. — 95983 A 192. — 96033 A 234. — 96034 A 257. — 96035 A 257. — 96054 A 257. — 96143 A 215. — 96144 A 234. — 96175 A 216. — 96176 A 234. — 96177 A 234.

96218. Bentall, E. E., Heybridge b. Maldon, Essex, Engl.; Vertr.: C. H. Knoop, Dresden. Sicherheitsvorrichtung für Häckselmaschinen zur Verhinderung des Bruches der Maschine beim Durchgang harter Fremdkörper. 10/2 97. — A 257.

96309. Mende, C., Repsch b. Ober-Glogau. Fangwagen für Raps- und andere Käfer. 8/4 97. — A 329.

(†) 96396. Grieger, A., Freiburg i/Schl. Säerad mit Saatkanälen als Ersatz der Reid'schen Säescheiben bei Säemaschinen mit durchlochtem Bodenschieber. 10/10 96. — A 278.

96397. Dall, J., u. J. Jensen, Kolding; Vertr.: D. Timar, Berlin N.W., Luisenstr. 27/28. Rübenerntemaschine mit federnden Gabeln. 29/11 96. — A 278.

(†) 96421. Cavatte, A., Baudonvilliers par Sandrupt, Meuse, Frankr.; Vertr.: F. Haßlacher, Frankfurt a/M. Pflug mit gegabeltem, zum Einspannen des Zugthieres dienendem Pflugbaum. 2/8 96. — A 278.

96422. Laaß, Hermann, & Co., Magdeburg-Neustadt. Zwangläufig bewegte Reinigungsvorrichtung für die Hebmesser an Rübenhebern. 13/12 96. — A 278.

96423. Gábor, A., Naszód; Vertr.: M. J. Hahlo, Berlin N.W., Luisenstr. 39. Handsäegeräth (Säestock). 17/12 96. — A 278.

(†) 96424. von Kaali, B. N., Kraszna, Ung.; Vertr.: F. C. Glaser u. L. Glaser, Berlin S.W., Lindenstr. 80. Köpfvorrichtung für Rübenerntemaschinen. 2/6 97. — A 279.

96425. Lins, C., Berlin N.W., Thurmstr. 44. Schubfäerad-Gehäuse mit um eine gemeinsame Achse drehbarem Boden und Schlußmuffenkamm; Zus. z. Pat. 96054. 25/6 97. — A 294.

96488. Wall, W., Obdrup b/Satrup, Angeln, Schleswig. Selbstthätige Futterzuführung für Pferdekrippen. 9/6 97. — A 313.

96521. Grünberger, Dr F., E. Tüchler u. J. Klein, Wien; Vertr.: A. Schmidt, Berlin, Friedrichstr. 138. Einschiebstollen, -Griffe oder -Stege mit elastischer Einlage für Hufeisen. 4/4 97. — A 329.

96522. Urban, A., Görbersdorf, Kr. Waldenburg i/Schl. Sensendengelmaschine mit mehreren neben einander arbeitenden Hämmern. 13/4 97. — A 295.

96645. Hoffmann sen., F. H., Karsdorf b/Pegau i/S., Zeitzerstr. 33. Siebtrommel mit Windrad zur Getreidereinigung. 24/1 97. — A 295.

96646. Melichar, F., Brandeis a/E., Böhm.; Vertr.: P. Rückert, Gera, Reuß. Vorrathskasten für Löffel- und Schöpfradsäemaschinen. 21/2 97. — A 313.

(†) 96647. Melichar, F., Brandeis a/E., Böhm.; Vertr.: P. Rückert, Gera, Reuß. Steuervorrichtung für Säemaschinen. 5/5 97. — A 329.

96648. Urban, A., Görbersdorf, Kr. Waldenburg i/Schl. Sensendengelmaschine mit selbstthätigem, während des Betriebes verstellbarem Vorschub. 15/6 97. — A 295.

(†) 96649. Göhlers, Wilhelm, Wittwe, Freiberg i/S. Geräth zum Durchschneiden der Wurzeln zwischen Pflanzenreihen, um späteres Umpflanzen zu erleichtern. 6/7 97. — A 329.

(†) 96650. Uhl jr, J., Ravensburg, Württ. Getreide-Reinigungsmaschine. 14/8 97. — A 295.

96750. Shiels, Dr A., Glasgow; Vertr.: H. Pataky u. W. Pataky, Berlin N.W., Luisenstr. 25. Durch abwechselnd fallende und steigende Luftsaugung wirkende Melkmaschine. 25/6 95. — A 445.

96751. Schöpke, A., Liebe b/Trebel, Hann. Pflug mit einem feststehenden und einem verstellbaren Streichbrett zur Herstellung verschieden breiter Furchen. 30/3 97. — A 329.

(†) 96837. Steen, P., u. C. Rödiger, Hamburg, Marthastr. 27. Vorrichtung zum selbstthätigen Füttern von Vieh. 2/5 97. — A 412.

(†) 96838. Koch, M., Niedergurig b/Bautzen. Hackmaschine mit durch Kurbelwelle hin- und hergeschobenen Hackmessern. 8/5 97. — A 384.

96839. Ritter, H., Oberschütz b. Hümme b/Kassel. Pferdestriegel. 13/5 97. — A 351.

96840. Rheinische Gesellschaft für Metallindustrie Greve, Herzberg, & Co., Köln, Pfälzerstr. 55. Fliegenfänger. 10/6 97. — A 368.

(†) 96917. Sonnenschein, H., Berlin, Neues Thor 5. Leine mit durch Schnurzug zu öffnendem Schloß für Jagdhunde. 21/4 97. — A 351.

97001. Lasetzki, R., Bledzianow b/Antonin, Posen. Bienenwohnung. 16/6 97. — A 412.

97050. de Kesel, K., Brüssel; Vertr.: F. C. Glaser u. L. Glaser, Berlin S.W., Lindenstr. 80. Bienenwohnung. 5/11 96. — A 430.

97051. Wunderlich, H., Borna b/Dresden. Einrichtung an Futtertrögen zum Befestigen derselben an der Wagendeichsel. 3/6 97. — A 430.

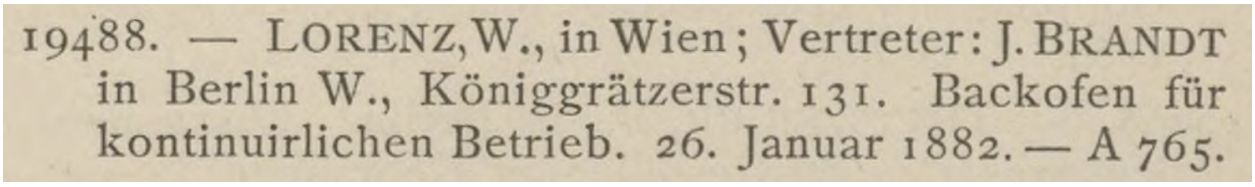
97089. Nadányi, N., Kovácsi puszta, Biharer Comitat, Ung.; Vertr.: G. Dedreux, München. Geräth zum Schneiden der Weinreben beim Veredeln. 20/2 97. — A 396.

97090. Hovell, Th. M., Boreham Holt, Elstree, Engl.; Vertr.: C. Gronert u. P. Kühne, Berlin N.W., Luisen-

Notes: A representative page from our 9,562 image scans (volume 1898, page 73). Please note how the patent entries at the top left and bottom right span multiple pages, as well as the single patent entry spanning both columns. Also note how much information we do not want to include, such as the running footer or the list of patent IDs that appears after the introduction of a new technological class. The introduction of a new technological class occurs randomly across both content columns. Moreover, many pages do not even introduce a new technological class.

However, there are also several fine-grained differences across our image corpus. Most notably, the general font type changes from Roman to Gothic in 1894. Within the Gothic period, the volumes from 1894 to 1911 use the Unger typeface, while those from 1911 to 1918 are set in Breitkopf. Later volumes introduce technological subclasses (e.g., "17a", "89k") as the number of patent registrations increased. The classifications become even more granular as patent numbers are preceded by enumerations (e.g., "17. 287909") to indicate the patent group. In addition, there are many minor variations across volumes. For example, the format of the registration date varies and some patent numbers are preceded or followed by a dagger symbol "(†)", denoting patents that had already expired before the volume was published.

FIGURE 2
A PATENT ENTRY


19488. — LORENZ, W., in Wien; Vertreter: J. BRANDT in Berlin W., Königgrätzerstr. 131. Backofen für kontinuirlichen Betrieb. 26. Januar 1882. — A 765.

Notes: A representative patent entry depicted on page 1 in volume 1882. This patent lists a legal representative because it was registered by an Austrian. It does not include a priority claim, which would typically appear at the very end.

5 Multimodal LLMs for Dataset Construction from Image Scans

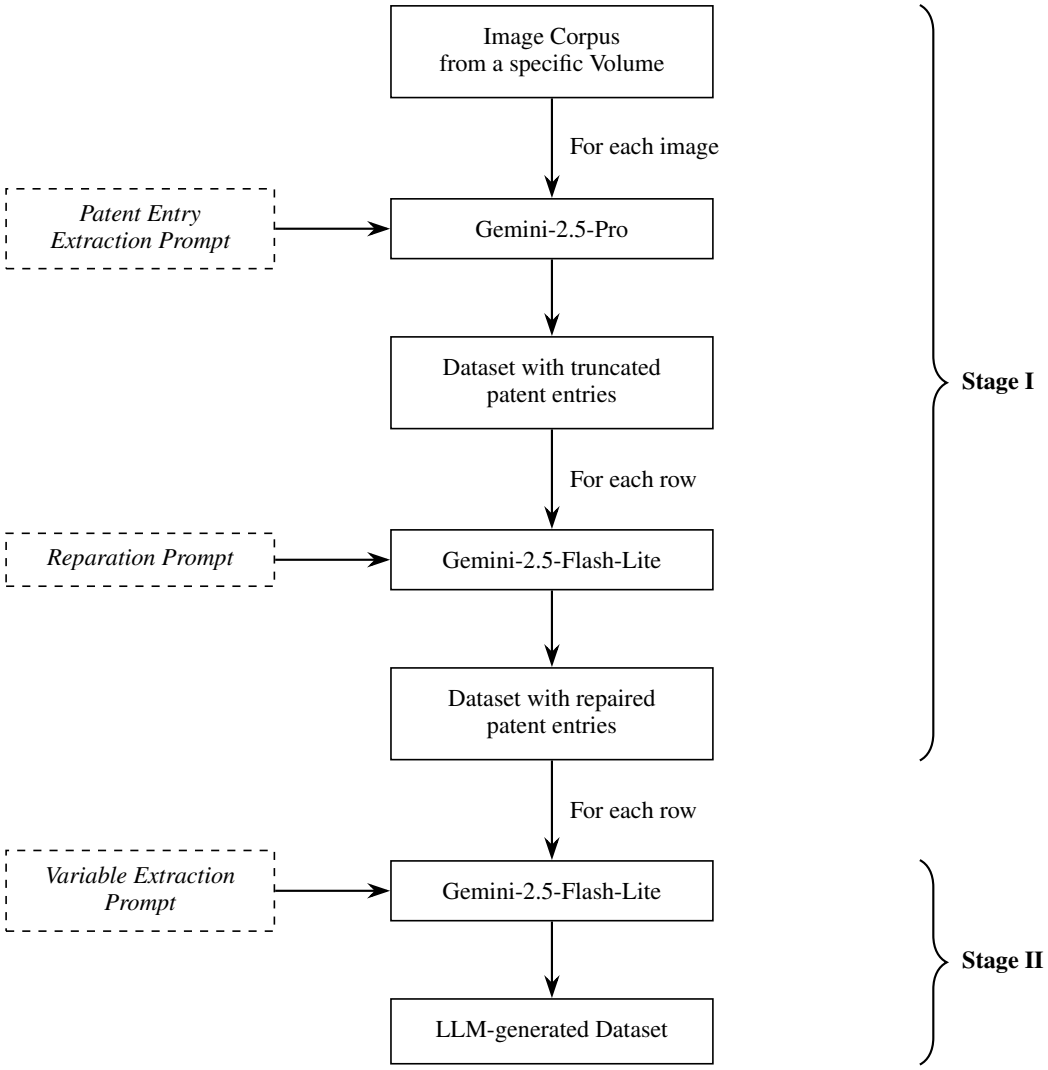
An overview of our LLM-based data pipeline to construct datasets from image scans is depicted in Figure 3. In the first stage, we pair each page (e.g., Figure 1) with the same prompt and send all prompt-page combinations independently to a multimodal LLM. As output we receive one JSON file per page, containing all patent entries. For a given volume, we concatenate all JSON files and convert them into one large CSV file. At the end of the first stage, our pipeline repairs patent entries that span across pages and columns and returns a dataset in which every observation contains one complete patent entry. In the second stage, we pair each extracted patent entry with a new prompt and send these prompt-entry combinations to another LLM. We receive the extracted variables we specified in our prompt, which are then appended to the corresponding patent entry in our dataset. The following subsections describe the technical details: why we chose the Gemini model family to power our pipeline, how we handled technological classes, repaired patent entries spanning multiple columns or pages, adapted the pipeline to volumes with a special layout, and conducted the manual validation and cleaning of our LLM-generated patent dataset.

5.1 Model Selection

We exclusively use models from the Gemini 2.5 family to power our LLM-based data pipeline (Comanici et al., 2025).[2] Critics may argue that we are using proprietary models offered by a private company. Open-source LLMs do exist, however, initial experiments clearly showed that they are far from capable of extracting the desired information from our image scans. Even if capable open-source alternatives had existed, they would have required access to expensive GPU hardware, which we did not have. Proprietary models from other frontier labs, such as Anthropic or OpenAI, also failed to deliver promising results. Therefore, at the time of coding, Gemini-2.5-Pro was the only model with promising visual capabilities for extracting information from our image corpus. In the end, the majority of economic historians is most likely indifferent about the LLM powering their data pipeline—as long as it quickly produces a high-quality dataset at a cheap price.

[2]We do not endorse any mentioned products, services, or organizations, and do not provide legal or financial advice.

FIGURE 3
OUR LLM-BASED DATA PIPELINE



Notes: This flowchart represents our LLM-based data pipeline for a given volume. The output is an LLM-generated dataset. After manual data cleaning, we merge all 41 LLM-generated datasets to construct the complete patent dataset including all German patents from 1877 until 1918. Dashed boxes depict our carefully refined prompts, which are shown in Appendix A. Throughout our pipeline, the temperature parameter is set to 0.0 for all model invocations.

5.2 Stage I: Patent Entry Extraction from Image Scans

Our first stage builds upon work by Greif et al. (2025). Initially, the PDF for a given volume is split into PNGs. Each page is paired with the same prompt (Figure A.1) and sent to Gemini-2.5-Pro via the Application Programming Interface (API).[3] This is done independently for all PNGs at the same time. The prompt is crucial to properly extract the desired information from the archival image scan. We carefully instruct the model to extract all patent entries and technological classes from top to bottom, starting with the column on the left before proceeding with the column on the right-hand side. For each page, we receive a JSON object that contains all patent entries and technological classes in sequential reading order. We retry pages whose API call to the LLM failed until they succeed. Patent entries are always preceded by the key *entry*, whereas technological classes are preceded by *category*. We concatenate all JSON files in document order and keep track of the page number. Consequently, we know that all patent entries appearing between two technological classes must belong to the technological class that precedes them. This allows us to convert the concatenated JSON file into a CSV with three variables: *page*, *entry*, and *category*.

Processing every page independently causes one significant issue. Some patent entries begin at the very bottom of the right-hand column and continue at the top-left on the subsequent page. Similarly, entries occasionally span across both columns on a given image, making it more challenging for Gemini-2.5-Pro to correctly extract the complete text. For this reason, some patent entries are truncated after they were extracted from the image scans. We address this issue by sending each entry to Gemini-2.5-Flash-Lite with a prompt (Figure A.2) that checks whether an entry is truncated (1) or not (0). We chose Gemini-2.5-Flash-Lite because it is much cheaper than Gemini-2.5-Pro while being as capable of solving this issue. Finally, we merge truncated entries that span across two pages or columns on the primary source material.

One may ask why we did not just send all pages in one PDF to Gemini-2.5-Pro in order to avoid the issue of entries exceeding a single page. First, the retrieval and reasoning performance of multimodal LLMs declines when more images are provided (Wu et al., 2025). Second, the maximum output window of Gemini-2.5-Pro is capped. Therefore, we would have to send all volumes in multiple chunks as they typically consist of hundreds of pages, resulting in the same patent entry continuation problem at the first and last pages of each chunk. Third, by processing each page independently, we do know exactly on which PNG a patent entry is located, facilitating manual data cleaning later on.

5.3 Stage II: Variable Extraction from Patent Entries

After the first stage of our LLM-based data pipeline, we now have a variable *entry* in which each row contains a complete patent entry. Conditional on each entry, we extract the variables *patent_id*, *assignee*, *location*, *title*, and *date* using Gemini-2.5-Flash-Lite. We instruct the model to extract the specified variables, provide some examples, and append the actual patent entry at the bottom of the prompt (Figure A.3). In each API call, the model only processes a single, isolated patent entry that was extracted in the first stage. If an API call fails, it is automatically retried up to ten times. Gemini-2.5-Flash-Lite returns a JSON object with keys for the desired variables. If a variable is not present in the patent entry, we instruct the model to return *NaN*. At the end of Stage II, each of the 41 volumes was transformed into a dataset containing the following variables: *page*, *entry*, *category*, *patent_id*, *assignee*, *location*, *title*, and *date*.

5.4 Special Volumes with Different Layout

The first two volumes (1877–8 and 1879) have a slightly different layout than the volumes from 1880 to 1918. The very first volume combines 1877 and 1878 as the Reich Patent Office was established on July 1, 1877. The major difference for these two volumes is that the ID is located at the end of each patent entry and is always preceded by "P. R.". Moreover, the combined volume 1877–8 does not contain information on the location of the assignees. Our pipeline allows us to easily adjust the prompts. After experimenting with various prompts on the benchmarking dataset, we amended the prompts across all stages of our pipeline to construct the datasets for the years 1877–8 and 1879. The prompts for these two volumes can be found in Appendix B.

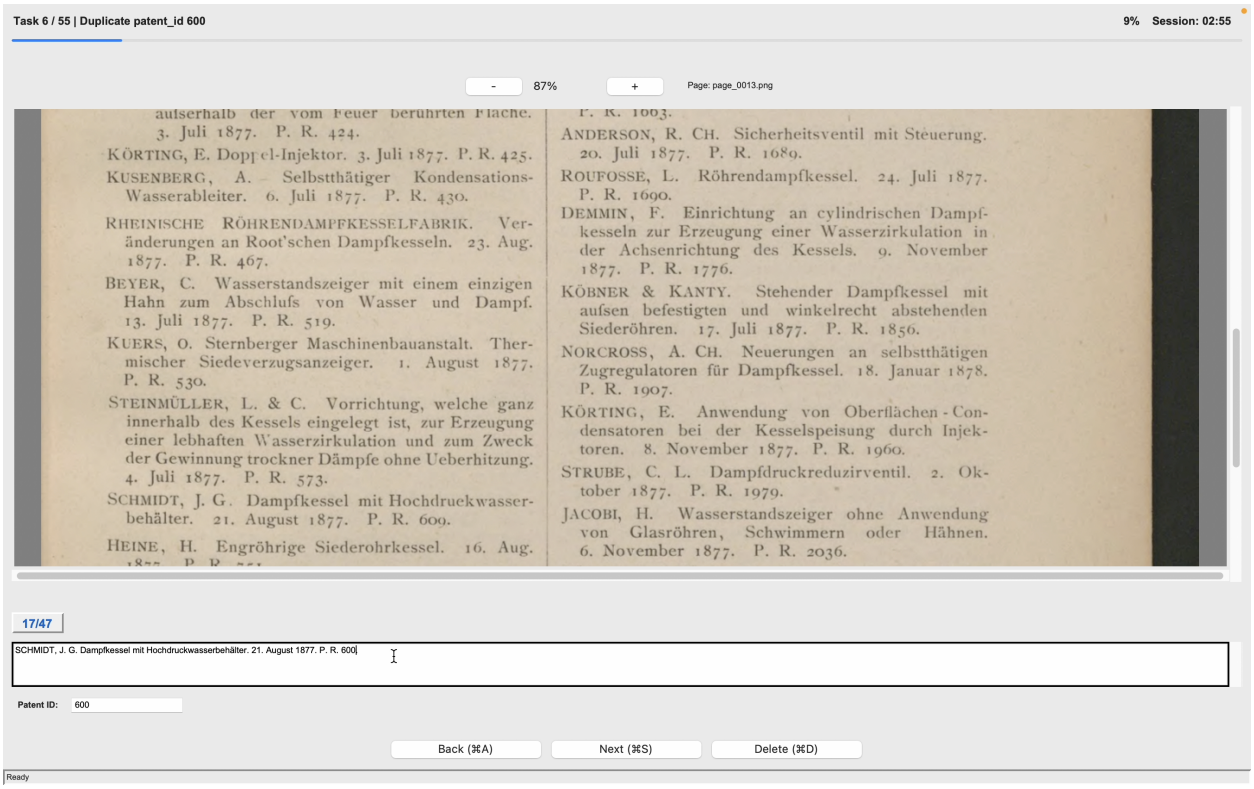
[3] An API is the standard way to send input to an LLM hosted by a commercial provider in its data centers, which then returns the model's output.

5.5 Manual Data Cleaning

After our LLM-based data pipeline constructed 41 datasets—one for each of the physical volumes—our research assistants manually validated and cleaned a small number of observations. We created auxiliary variables throughout our pipeline to flag truncated entries that were merged, entries for which the API call did not succeed, and samples for which no patent ID was found. Using these auxiliary variables, our research assistants filtered each of our 41 datasets to check, repair, and delete these observations.

Afterward, we exploited the structure of the primary sources to ensure the quality of each of the 41 datasets. Patents were recorded starting with one-unit increments from patent number 1. The table of contents indicates the first and last patent number of the corresponding volume, allowing us to check for duplicates and resolve entries with patent numbers that are below or above the volume's range.[4] Using LLM-assisted coding tools, we quickly constructed a visual interface for our research assistants (Figure 4). For example, if a patent ID was marked as duplicate, the interface rendered the corresponding page at the top, while the current values for the entry and ID were editable below. Our research assistants then only had to check the image scan and update the patent entry and ID fields, if applicable. Throughout this validation process, we also noticed that the primary source itself contained errors. For example, the patent clerks occasionally made mistakes, resulting in patent entries with duplicate IDs.[5]

FIGURE 4
INTERFACE FOR ACCELERATED MANUAL DATA CLEANING



Notes: We created this interface using LLM-assisted coding tools. Our research assistants used it to manually clean and validate patent entries with duplicate ID values or IDs that fall outside the volume's range. In the example, the lower part of the "9" in patent ID "609" is faintly printed, which caused Gemini-2.5-Pro to extract it as "600". Because every patent ID is unique, this discrepancy was flagged. Our research assistants then corrected these mistakes by verifying the outputs with their own eyes, whenever applicable. Our interface accelerated this process as our research assistants could make an edit and just had to click "Next".

[4] Due to World War I, around 4,000 patent numbers were not entered into the patent register for the volumes 1917 and 1918.

[5] We did not correct these errors made by the employees of the Reich Patent Office and did not assign new, ahistorical patent numbers. Our dataset therefore occasionally contains two different patents with the same patent number.

The structure of our primary sources also helped us to identify observations in which the technological class was transcribed incorrectly. In a given volume, all patents were listed sequentially with respect to their technological class. For example, class "2" must follow class '1". Similarly, subclass "17e" must come after "17d". By highlighting inconsistencies among these sequences, our research assistants manually corrected the *category* variable.

The deviating content of some volumes forced us to auto-delete patent entries in bulk. For the year 1888, we were only able to track down a composite volume that contains all patents that were registered between July 1, 1878, and December 31, 1888, and were still active. Therefore, this volume contains patent entries from preceding volumes, which we removed. Similarly, the volume for year 1879 includes patent entries that were still active and listed in the first volume covering the years 1877 and 1878. These were also removed.

After we conducted the manual data cleaning described above for all 41 LLM-generated datasets, we merged them into a single dataset and prepended the variables *global_id*, *book*, and *book_id* to provide high-level metadata.

6 Benchmarking

The benchmarking dataset serves two intertwined purposes. Of course, it allows us to evaluate the quality of our dataset that is produced by our LLM-based data pipeline on a small but representative sample. However, the dataset quality is highly dependent on the prompts we select each time our pipeline employs an LLM. Thus, the benchmarking dataset also enables us to find prompts that maximize the accuracy of the LLM-generated dataset. As we optimize the performance of our LLM-based data pipeline on a representative corpus of 41 randomly selected images, we argue that the evaluation metrics reported on this sample generalize to our full image corpus.

6.1 Hallucinations

LLMs have often been caught hallucinating as they can fabricate convincing but incorrect information. In order to assess whether this problem occurs in our task, we had to build a human-made benchmarking dataset with the help of research assistance, which can be used to evaluate the output of multimodal LLMs.

There are several types of hallucinations. In our dataset construction task, we can only encounter "input-conflicting hallucinations, where LLMs generate content that deviates from the source input provided by users" (Zhang et al., 2025, p. 5). More precisely, we define a hallucination as any deviation of the *LLM-generated* dataset with respect to the *perfect* benchmarking dataset.

For our research project, not all hallucinations are equally concerning. Falsely recognizing characters or using modern instead of historical letters may be treated as minor hallucinations. These are negligible when it comes to analyzing the historical mass data. More worrisome are moderate hallucinations, such as when the multimodal LLM extracts a technological class header as a patent entry or when the LLM transcribes a full word with modern instead of historical spelling. Major hallucinations are very concerning regarding the dataset quality and its reliability for downstream econometric analysis. This includes completely false or misinterpreted transcriptions of the patent entry and, even worse, invented patent entries that do not appear on the image scan and therefore do not exist.

6.2 Benchmarking Datasets: Student-Constructed and Perfect

We randomly selected 41 image scans, one for each calendar year of the annually published patent register, distributed them across two research assistants, and carefully instructed them on how to create the dataset in Excel. In what follows, we refer to the output they initially produced as the *student-constructed* dataset. This process of constructing datasets from archival image scans with the help of research assistants was standard practice for decades in economic history research. Consequently, we are confident that the *student-constructed* dataset approximates the average quality of existing economic history datasets.

After the research assistants produced the *student-constructed* dataset, we generated the corresponding dataset using our LLM-based data pipeline on the 41 sampled images. We then compared the *LLM-generated* dataset with the *student-constructed* dataset to assess the accuracy of our pipeline. We noticed major differences between the two datasets. Initially, we suspected that Gemini-2.5-Pro hallucinated patent entries, but we quickly realized that the

research assistants had accidentally skipped some entries. We also observed several transposition errors in which the students extracted "299187" as the patent number, but the LLM correctly wrote "299178". In addition, we noticed many typos, such as repeated or omitted characters. For this reason, we asked the research assistants to correct their initial *student-constructed* benchmarking dataset until it was *perfect*. To the best of our knowledge, the *perfect* dataset does not contain any remaining errors.

6.3 LLM-assisted Iterative Prompt Refinement on the Benchmarking Dataset

Our approach to finding optimal prompts throughout our pipeline resembles contemporaneous work by Bäcker-Peral et al. (2025) and Vafaie et al. (2025). We iteratively refined the prompts across our whole pipeline with the aim of generating a dataset that matches the *perfect* benchmarking dataset as accurately as possible. After each iteration, we examined our selected evaluation metrics and inspected side-by-side comparisons of the *LLM-generated* and *perfect* datasets. This visual inspection revealed the shortcomings of our current prompts. We described these inaccuracies (e.g., for an early prompt, locations were extracted with the prepositions "in" when creating the location variable; "in Berlin") and instructed our LLM-assisted coding tool to improve our prompts by taking these issues into account. We carried out several iterations of this LLM-assisted prompt refinement technique. Most likely, there exist other prompts that would yield even higher values on our evaluation metrics. However, after visual inspection, we concluded that our prompts were strong enough to ensure high data quality on our full image corpus.

We wrote the initial prompts by infusing our historical domain knowledge about the primary source. As a general rule, we started a prompt by assigning a role to the LLM (e.g., "You are a specialist data extraction AI."), followed by a brief outline of the general task. We then provided very detailed and explicit instructions, after which we demonstrated some examples with their desired JSON output, if applicable. At the very end, we frequently instructed the models to provide only JSON output and nothing else to avoid unnecessary explanations, which would interfere with the subsequent processing steps in our pipeline, such as concatenating JSON objects or integrating them into the dataset.

6.4 Character Error Rate

Before benchmarking the Stage I and Stage II of our pipeline, we evaluated the transcription capabilities of Gemini-2.5-Pro by focusing exclusively on the patent entries. Nevertheless, this required performing the first part of Stage I, namely sending the 41 benchmarking images to the model. For each volume, we concatenate all *LLM-generated* patent entries into a single text file to measure the transcription accuracy. We do the same for the *student-constructed* and *perfect* benchmarking datasets.

We then use the character error rate (CER), a standard metric from the field of OCR, to quantify how different two text documents are at the character level. The CER is defined as follows:

$$\text{CER} = \frac{\text{Levenshtein Distance}}{N} = \frac{S + D + I}{N} \tag{1}$$

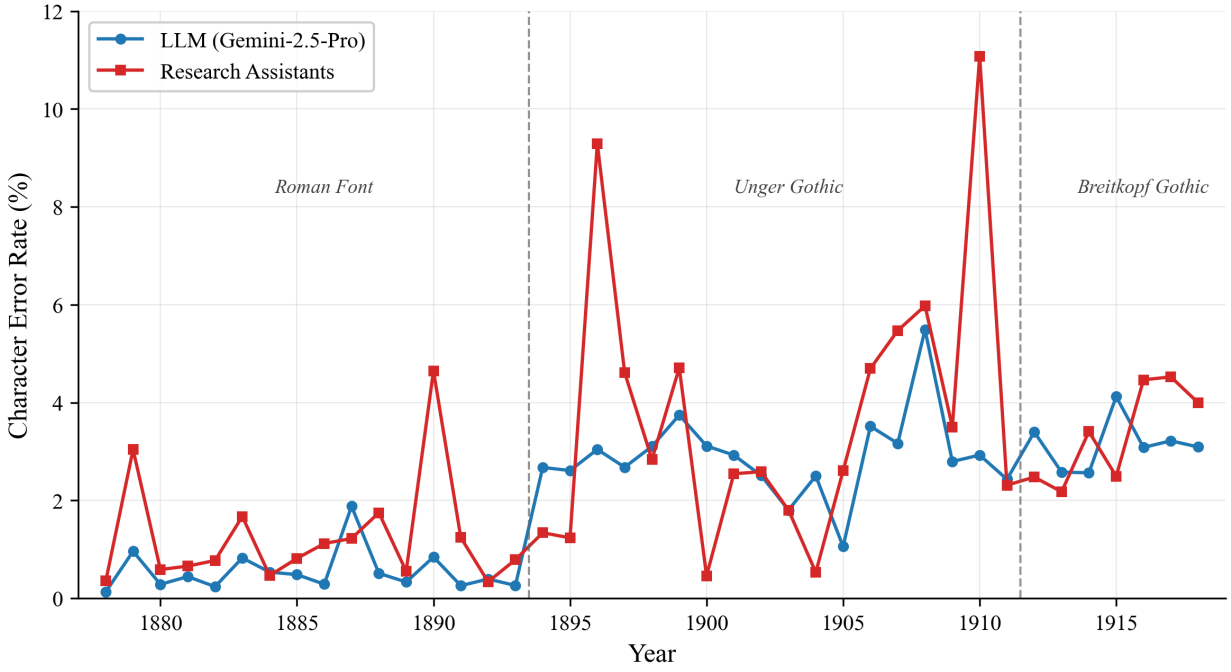
where the Levenshtein distance is defined as the sum of the number of substitutions S, the number of deletions D, and the number of insertions I. N is the number of characters in the *perfect* text file. Intuitively, the CER is the minimal number of keystrokes needed to turn the *LLM-generated* patent entry transcriptions into the *perfect* text file (or vice versa), normalized by the length of the *perfect* file.

Figure 5 reports the CER distance both between the *student-constructed* and *perfect* transcriptions and between the *LLM-generated* and *perfect* transcriptions across all 41 volumes. Gemini-2.5-Pro produces transcriptions that are closer to the *perfect* version than the efforts of our research assistants for 27 of the 41 text files. There is one file in which both yield the same CER, while the research assistants are more accurate on the other 13 files. The CER clearly increases for the Gothic fonts, but this appears to hold true for our research assistants as well, who told us that they had to rely on a Gothic reading aid. A closer inspection shows that Gemini-2.5-Pro mainly struggles with the historical long s and occasionally with capitalization.[6]

[6]We provide an accompanying website that presents all benchmarking results in a visual format, allowing readers to assess the quality of the dataset for themselves. See historymind.ai.

The fact that the CER on our patent entries tends to be lower than that of our research assistants is highly reassuring with regard to severe hallucinations. For example, a single hallucinated patent entry would increase the CER by 2–3% at the volume level. Nevertheless, we found five notable hallucinations that we wanted to discuss to be fully transparent about the quality of our LLM-generated dataset. The most severe hallucination we detected occurred with our benchmarking image from volume 1878. Gemini-2.5-Pro partially fabricated non-existent parts of the title of the patent with ID 7763. Moreover, the last patent entry on the benchmarking image in volume 1907 was completed by the model with four words that were not depicted in the image. The model also failed to correctly extract the first patent entry in our benchmarking image from volume 1915, as part of the beginning is missing. For the image from volume 1908, the transcription of the patent with ID 199989 is missing an additional priority claim that follows the application date. Finally, on the benchmarking image from volume 1913, the patent entry with ID 259447 reads "Versteinung", whereas the LLM incorrectly extracted "Verkoksung". All of these examples demonstrate one important point: while Gemini-2.5-Pro tends to perform better than our research assistants at transcribing patent entries, the resulting dataset based on our double-column primary sources remains imperfect.

FIGURE 5
CHARACTER ERROR RATE BY YEARLY VOLUME: RESEARCH ASSISTANTS VS LLM



Notes: We sampled one page from each of the 41 annual volumes. Each data point is the CER computed over all *LLM-generated* (blue) or *student-constructed* (red) patent entry transcriptions with respect to the *perfect* ones. We did not normalize the transcriptions because we wanted to preserve their historical accuracy. This is an unusual choice and leads to higher CER values, but allows us to see more clearly how close our research assistants and Gemini-2.5-Pro come to the *perfect* transcriptions that are actually depicted on the archival image scans.

6.5 Stage I: Patent Entry Extraction from Image Scans

While the CER provides tentative evidence that Gemini-2.5-Pro can extract and transcribe patent entries more accurately than our human research assistants, we also need to assess how well our LLM-based pipeline can preserve structure and repair patent entries.

We perform one-to-one matching between the datasets containing the *LLM-generated* and *perfect* entries. To allow for small differences caused by misidentified characters, we employ fuzzy string matching. Two patent entries match if they are at least 90% identical at the character level, which approximately corresponds to allowing errors in no more than one

tenth of the characters.[7] We perform this matching before and after repairing truncated entries. In this benchmarking exercise, truncated entries can only be repaired when an entry spans columns, because we sampled one page from each volume. This is conceptually identical to truncated entries spanning two subsequent pages, providing a solid testing ground for evaluating the reliability of our reparation procedure.

Table 1 depicts the benchmarking results. Directly after the patent entry extraction, 1,360 out of 1,403 *LLM-generated* entries match one of the 1,385 *perfect* entries. 18 of the 41 benchmarking image samples contain a patent entry that spans both columns. Initially, both parts of these spanning entries are extracted as two separate patent entries, which explains why Gemini-2.5-Pro extracted more entries than actually exist. After repairing these spanning entries, the *LLM-generated* benchmarking dataset also consists of 1,385 entries, of which 1,378 have a matching counterpart in the *perfect* benchmarking dataset. Five of the seven unmatched entries are not paired due to the hallucinations described in Section 6.4. The remaining two contain a high proportion of the historical long s character but are otherwise largely correct.

Table 1:
STAGE I BENCHMARKING RESULTS

	Extraction (Gemini-2.5-Pro)	**Reparation** (Gemini-2.5-Flash-Lite)	**Student-Constructed**
Perfect Entries	1,385	1,385	1,385
Extracted Entries	1,403	1,385	1,373
Matched Entries	1,360	1,378	1,358
% *Perfect* Matched	98.19%	99.49%	98.05%
% Extracted Matched	96.94%	99.49%	98.91%

Notes: The *perfect* benchmarking dataset consists of 1,385 patent entries. Column "Extraction" refers to the initial part of the first stage where patent entries are extracted from image scans using Gemini-2.5-Pro. Column "Reparation" reports performance after repairing truncated entries using Gemini-2.5-Flash-Lite. The third column displays the performance of our research assistants, which is best compared to the "Reparation" column. Matches were determined using the rapidfuzz library's normalized Levenshtein similarity metric with the threshold set to 0.90. The reported percentages are also known as Recall (% *Perfect* Matched) and Precision (% Extracted Matched).

6.6 Stage II: Variable Extraction from Patent Entries

To benchmark the second stage, we evaluated how accurately Gemini-2.5-Flash-Lite extracts the desired variables from the subsets of patent entries that matched in the previous step (Section 6.5). Within those two subsets (*LLM-generated* and *student-constructed*), we extract the desired five variables (*patent_id*, *assignee*, *location*, *title*, and *date*) for each patent entry and compare them with the corresponding variables in the *perfect* benchmarking dataset.

To determine whether two variable cells match, we apply fuzzy string matching. By choosing the conservative threshold of 90%, we allow for transcription errors in longer fields, such as patent titles, while still requiring exact correspondence for short fields such as the six-digit patent ID or date.

In Stage I, 1,378 entries match between the *LLM-generated* and *perfect* benchmarking dataset, yielding 6,890 variable cells, of which 6,550 match in Stage II of our benchmarking exercise, a percentage of 95.07% (Table 2). The shorter patent ID and date field have an accuracy of 99.49% and 98.91%, respectively. The accuracy is lower for assignees (92.16%), locations (91.36%), and patent titles (93.40%), which is expected due to transcription errors from the long s and capitalization that patent IDs and dates do not usually have.

Notably, the *student-constructed* benchmarking dataset only had 1,358 matching entries with respect to the *perfect* dataset because the students completely overlooked some patent entries or incorrectly repeated patent titles from previous entries, among other things. Therefore, the *student-constructed* dataset is evaluated on a smaller subset, making a direct comparison with the *LLM-generated* dataset more difficult. Nevertheless, even on this reduced subset,

[7] Greedy one-to-one fuzzy matching was performed using the rapidfuzz library's normalized Levenshtein similarity metric with the threshold set to 0.90.

the research assistants were less accurate in extracting patent IDs, locations, and dates. They were better in extracting assignees and titles. This pattern is explained by the fact that we explicitly instructed our research assistant to use the historical long s, which Gemini-2.5-Pro failed to reproduce.

Some economic historians may argue that the reported variable-level accuracies on the *LLM-generated* dataset are insufficient, or that the remaining errors are systematically biased. For this reason, we closely examined the non-matching variable cells.[8] For the assignee field, Gemini-2.5-Flash-Lite sometimes incorrectly included the occupation, which occasionally followed an individual name. Furthermore, Gemini-2.5-Flash-Lite has consistently separated each of the names of three or more patent holders with "und" (German for "and"), while the perfect benchmarking dataset follows a comma-separated listing convention, with "und" only appearing before the last name. The *LLM-generated* locations often include a subsequent end point without spaces, which is not present in the *perfect* benchmarking dataset (e.g., *LLM-generated* "Berlin." and *perfect* "Berlin" do not match).[9] Moreover, many locations contain the historical long s. These single character deviations can cause a non-match as many of the extracted locations are short (e.g., "Dresden" does not match its counterpart with the historical long s). Based on our benchmarking results, we conclude that our *LLM-generated* patent dataset is of sufficiently high quality for use in economic history research.

Table 2:
STAGE II BENCHMARKING RESULTS

	Gemini-2.5-Flash-Lite			**Student-constructed**		
	Total	Matched	Match Rate	Total	Matched	Match Rate
Total Cells	6,890	6,550	95.07%	6,790	6,503	95.77%
By Variable						
Patent ID	1,378	1,371	99.49%	1,358	1,329	97.86%
Assignee	1,378	1,270	92.16%	1,358	1,289	94.92%
Location	1,378	1,259	91.36%	1,358	1,231	90.65%
Title	1,378	1,287	93.40%	1,358	1,316	96.91%
Date	1,378	1,363	98.91%	1,358	1,338	98.53%

Notes: This table reports variable extraction performance comparing *LLM-generated* and *student-constructed* variables against the variables from the *perfect* benchmarking dataset. Based on the benchmarking results in Table 1, we retain 1,378 *LLM-generated* and 1,358 *student-constructed* entries, resulting in 6,890 and 6,790 variable cells. For each patent entry, we extract five variables (*patent_id*, *assignee*, *location*, *title*, and *date*) and compare them with the variable values in the *perfect* benchmarking datasets. Matches between two cells were determined using the rapidfuzz library's normalized Levenshtein similarity metric with the threshold set to 0.90.

7 The Economics of Constructing Datasets from Image Scans Using Multimodal LLMs

Until now, economic historians around the world have been constructing datasets from primary sources by hand. The task of extracting information from archival sources is typically conducted by research assistants, PhD students, and sometimes even outsourced to specialized companies. This process is very time-consuming and costly. When we constructed our benchmarking datasets, we asked our research assistants to track their time. On average, they worked roughly two hours per page to produce the *student-constructed* dataset, including the full patent entries as well as the five desired variables. Creating the *perfect* benchmarking dataset required an additional two hours per page. As our image corpus corpus consists of 9,562 pages, it would have taken about 19,124 hours of human labor to construct the patent dataset using the standard practices in economic history. By contrary, our LLM-based data pipeline can generate the full dataset within a day. If we assume a German minimum wage of EUR 12.82, the construction of the patent dataset would have cost at least EUR 245,269. By contrast, the LLM-based data pipeline cost EUR 1,196 (Table 3). Consequently, multimodal LLMs enabled us to construct the patent dataset more than 795 times faster and 205 times

[8] All variable extraction benchmarking results are visually available on our paper website at historymind.ai.

[9] Further refining our prompts may have resolved these issues.

cheaper compared to the standard practice of constructing datasets in economic history.[10] The superiority of multimodal LLMs for historical data construction becomes even more pronounced by the fact that the pipeline yielded higher data quality than we would expect from our research assistants.

The primary cost driver in our LLM-based data pipeline is the model selection. To ensure the highest data quality, we could have exclusively used the most capable model of the Gemini family, Gemini-2.5-Pro at the time of coding, across all points where LLMs are employed in our pipeline. However, this would have led to much higher costs with only slight quality improvements. For this reason, we chose the less capable but cheaper Gemini-2.5-Flash-Lite for repairing truncated entries and extracting variables from the previously extracted patent entries. The model exhibited satisfactory performance on the benchmarking dataset, so why should we use a more intelligent and expensive model if the results are already satisfactory? As the cost of LLM inference continues to fall quickly (Gundlach et al., 2025), researchers have to constantly reevaluate which models meet their requirements, depending on their digitization project. Once open-source models reach the performance of current frontier models, they may become reliable alternatives for dataset construction tasks.

Table 3: Cost Breakdown by Pipeline Stage

		$/1M Tokens		**Tokens Used**			
Stage	**Model**	Input	Output	Input	Output	**Cost ($)**	**%**
Stage I: Patent Entry Extraction from Image Scans							
Extraction	Gemini-2.5-Pro	$1.25	$10.00	24.1M	107.6M	1,105.9	92.4%
Reparation	Gemini-2.5-Flash-Lite	$0.10	$0.40	183.1M	0.3M	18.4	1.6%
Stage II: Variable Extraction from Patent Entries							
Extraction	Gemini-2.5-Flash-Lite	$0.10	$0.40	612.9M	26.6M	71.9	6.0%
Total				**820.1M**	**134.5M**	**1,196.3**	**100%**

Notes: We tracked all costs within our LLM-based data pipeline using prices as of August 2025. We did not use batch processing, which would have reduced the overall cost by 50%. Output tokens include both thinking (reasoning) and output tokens. The major cost driver of our pipeline are the 107.6 million output tokens of Gemini-2.5-Pro. All token counts are rounded to one decimal place.

The cost to use an LLM is determined by the length of the input it receives and by the length of the output it produces. The length is measured in tokens. A text token can be a single word or a smaller unit of text. Analogously, images are converted to image tokens. In a single request to Gemini-2.5-Pro, the image scan and the prompt are tokenized and jointly represent the total input tokens. In general, input tokens are cheap, while output tokens are much more expensive. Due to these underlying economics, writing a very explicit prompt does not incur much cost. In contrast, Gemini-2.5-Pro's output tokens are very expensive, explaining why 92.4% of our budget was spent on extracting the patent entries from our image scans. Nevertheless, the patent entry extraction from the archival image scans is the foundation of the whole dataset quality, which is the reason why we wanted to use the best multimodal LLM available at the time of implementation. In addition to input and output tokens, Gemini-2.5-Pro and Gemini-2.5-Flash-Lite also generate thinking tokens. Upon a request, both models first "think and reason" before providing an answer. With increasing task difficulty, the models spend more resources on thinking tokens in order to outline and plan how to successfully accomplish the user request. These thinking tokens are charged at the same price as output tokens, which further adds to the high cost of our first stage.

[10] Research assistants hired in Baden-Württemberg, Germany currently earn approximately one euro above the minimum wage. Moreover, our pipeline could process all 9,562 image scans in a few seconds. However, we required roughly a single day because of the constraints of our personal computer and to avoid rate limit issues with the Google Gemini API. Consequently, our comparative calculations should be viewed as lower-bound estimates.

8 Conclusion

In this paper, we have made several contributions. First, we built and published an LLM-based data pipeline that made it possible to transform the image scans of the German historical patent register into a structured patent database that provides information about the patent holders and the content of all 306,070 patents of the German Empire. Second, we created and open-sourced a *student-constructed* and a *perfect* benchmarking dataset to evaluate the accuracy of multimodal LLMs on the task of historical dataset construction from image scans. Our benchmarking exercise provides tentative evidence that multimodal LLMs can generate datasets of higher quality than those constructed by human research assistants. While we can only be certain about this statement on our image corpus, there is strong reason to believe that this may also be the case for other and even more complex historical sources. Therefore, we encourage other economic historians to use LLM-assisted coding tools to adopt our pipeline to their image corpora in order to explore the general reliability of multimodal LLMs. Third, we open-sourced the full patent dataset, an invaluable source to study the history of innovation. In the remainder of this conclusion, we will discuss the potential impact of multimodal large language models on the field of economic history.

For a long time, economic history research suffered from a lack of large-scale microeconomic data. The advent of multimodal large language models might put an end to this problem. However, a systematic risk in deploying AI agents to construct large-scale datasets is the infeasibility of validating all data entries, which is exacerbated by our limited understanding of LLMs. The potential introduction of unnoticed hallucinations—or even worse—of systematic bias into LLM-generated datasets threatens the robustness of the subsequent econometric analysis. This can have harmful real-world consequences if insights based on biased datasets shape modern policy. Our LLM-generated dataset, for example, has 306,070 patent entries, each yielding the five extracted variables and the technological class, making a manual review of all 2,142,490 data cells practically impossible. Therefore, we have to rely on the results reported on our benchmarking dataset and assume that the performance on this representative subset of 41 pages will generalize to our full image corpus. The reliability of future patent-based historical research on the causes and consequences of German innovations is based on the accuracy of this assumption. Finally, we note that this new way of generating mass data departs from the standard practice in economic history, where close manual work has has helped economic historians to acquire a deep historical understanding and intuition regarding the primary source material.

Multimodal LLMs may soon allow economic historians to build datasets from image scans on demand by simply sending a PDF to a model with the instruction to output the dataset as CSV or JSON object.[11] Multimodal LLMs will also enable the research community to quickly replicate well-known economic history datasets and test whether the downstream analyses are robust. On-demand dataset construction from image scans may also steer the incentives of scholars toward open-sourcing the datasets they painstakingly created by hand over decades. These datasets were often kept private, as economic historians planned to use them for their own publications. However, on-demand dataset construction may also incentivize public digital archives to take down their collections of archival image scans, aiming to prevent others from downloading them and creating datasets on which they could then publish their own research.

Future research should benchmark how LLMs perform when constructing historical datasets based on handwritten fonts and low-resource languages, or when provided with a PDF containing multiple pages instead of processing each page independently. Furthermore, attempts should be made to implement text-as-data as image-as-data approaches. Why should economic historians work with editable text files when they can work directly with archival image scans depicting embedded text? In general, economic historians should strive to work with the closest representation of the primary source. For example, consider a paper that aims to determine the sentiment transmitted in every daily BBC television broadcast since 1936. Instead of relying on audio transcripts, economic historians should use multimodal LLMs that can directly process videos, which may capture more nuances in how information was conveyed to the

[11] During writing, Gemini-3-Pro-Preview was released in November 2025. This model shows even stronger visual abilities, and when prompted with "Create an economic history dataset" and given a multi-page PDF, it seems to output a near-perfect dataset while being able to handle truncated patent entries. Gemini-3-Flash-Preview was released shortly after in December 2025 and outperforms Gemini-2.5-Pro across almost all benchmarks while being more than three times cheaper. We initially developed the two-stage pipeline because, with Gemini-2.5-Pro, the best strategy was to keep dataset construction tightly controlled by breaking the task into smaller subproblems rather than asking the model to generate full datasets directly from one or multiple image scans. Therefore, we reserve the right to rebuild the dataset in the future with an even stronger model to increase dataset quality further and to rule out any remaining hallucinations.

nation. Economic history, like other social sciences and humanities, may enter an age of data abundance, in which economic historians can answer new questions simply by constructing new large-scale datasets from sources such as our patent statistics.

Multimodal LLMs may not change what is at the core of economic history, but they will accelerate the research cycle, as has already been the case in other domains (Bubeck et al., 2025). For example, identifying gaps in the literature and then proposing promising hypotheses has already yielded successful results in the biomedical domain (Gottweis et al., 2025). Moreover, there are ongoing efforts to build AI Scientists that automate data-driven scientific discovery (Mitchener et al., 2025). In economics, there is an emerging literature on AI agents to accelerate economic research (Korinek, 2025). However, such debates are almost absent in economic history, even though the field is even more susceptible to automated data-driven discovery, as all the answers are, in principle, based on the surviving documents and artifacts. While many questions about the future development and impact of AI remain uncertain, we are convinced that the nature of work in economic history will change. The tasks of research assistants will shift away from manually constructing historical datasets from primary sources to orchestrating LLM-based data pipelines with LLM-assisted coding tools and managing the underlying data infrastructure. Economic historians are no longer limited by manual dataset construction and will have to learn to navigate this abundance of data to filter out those research questions that promise new and important insights.

Acknowledgments

We are grateful for feedback from Philipp Ager, Stephen Broadberry, Alexander Donges, Gavin Greif, Leander Heldring, John Meyer, Sheilagh Ogilvie, the audience at the Mannheim PhD colloquium in September 2025, the audience at the launch event of the Oxford Digital Scholarship Society in November 2025, the audience at the Oxford Graduate Seminar in November 2025, and participants of the Oxford-Warwick-LSE (OWL) workshop in December 2025. Furthermore, we are highly thankful to Carsten Burhop, who lent us a large share of the physical volumes. We also thank Tünde Gottschling and Irene Schumm for their unwavering support in managing the scanning of all primary sources in the Digital Lab of the University of Mannheim. Finally, we extend thanks to our research assistants, in particular Julius Schöffel, for scanning the physical sources, constructing the benchmarking datasets, and manually cleaning the patent dataset. Our research was financially supported by the Heidelberg Academy of Sciences as part of its academy program "Finanz- und Unternehmensforschung in Langfristperspektive". Niclas Griesshaber is highly grateful for the DPhil funding provided by the Economic and Social Research Council through UK Research and Innovation under the Advanced Quantitative Methods Award.

AI Transparency Statement

The first draft of this paper was written by the authors without the assistance of any external tools. Afterward, LLMs were used to restructure logical flows, correct grammar and selectively deployed to improve readability by suggesting synonyms and alternative phrasing using the wording from our first draft. The code for our GitHub repository and the project website was generated completely by LLMs. In particular, we used the Cursor code editor, where we guided the coding agents with natural language to build our website and pipeline. We reviewed the LLM-generated code to the best of our ability to ensure a responsible use of AI when creating our codebase and website.

References

Abramitzky, R., Boustan, L., Eriksson, K., Feigenbaum, J. & Pérez, S. (2021), 'Automated linking of historical data', *Journal of Economic Literature*, 59(3), 865–918.

Albers, T. N. & Kappner, K. (2023), 'Perks and pitfalls of city directories as a micro-geographic data source', *Explorations in Economic History*, 87, 101476.

Bäcker-Peral, V., Meursault, V. & Severen, C. (2025), 'Can LLMs Credibly Transform the Creation of Panel Data from Diverse Historical Tables?', *arXiv preprint arXiv:2505.11599*.

Baek, I., Chang, H., Ryu, S. & Lee, H. (2025), 'How Do Large Vision-Language Models See Text in Image? Unveiling the Distinctive Role of OCR Heads', *arXiv preprint arXiv:2505.15865*.

Bartik, A., Gupta, A. & Milo, D. (2025), 'The costs of housing regulation: Evidence from generative regulatory measurement', *Available at SSRN 4627587*.

Bergeaud, A. & Verluise, C. (2024), 'A new dataset to study a century of innovation in Europe and in the US', *Research Policy*, 53(1), 104903.

Broadberry, S., Campbell, B. M., Klein, A., Overton, M. & Van Leeuwen, B. (2015), *British economic growth, 1270–1870*, Cambridge University Press.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P. et al. (2020), 'Language models are few-shot learners', *Advances in neural information processing systems*, 33, 1877–1901.

Bubeck, S., Coester, C., Eldan, R., Gowers, T., Lee, Y. T., Lupsasca, A., Sawhney, M., Scherrer, R. et al. (2025), 'Early science acceleration experiments with GPT-5', *arXiv preprint arXiv:2511.16072*.

Carlson, J. & Dell, M. (2025), 'A Unifying Framework for Robust and Efficient Inference with Unstructured Data', *arXiv preprint arXiv:2505.00282*.

Chyn, E., Haggag, K. & Maruthiah, C. (2025), Ideology in Government: Evidence from the Office of Indian Affairs and the Assimilation Era, Technical report, National Bureau of Economic Research.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O. et al. (2025), 'Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities', *arXiv preprint arXiv:2507.06261*.

Cox, G. W. (2019), 'Patent disclosure and England's early industrial revolution', *European Review of Economic History*, 24(3), 447–467.

Crosilla, G., Klic, L. & Colavizza, G. (2025), 'Benchmarking large language models for handwritten text recognition', *Journal of Documentation*, 81(7), 334–354.

Dell, M., Carlson, J., Bryan, T., Silcock, E., Arora, A., Shen, Z., D'Amico-Wong, L., Le, Q. et al. (2023), 'American stories: A large-scale structured text dataset of historical us newspapers', *Advances in Neural Information Processing Systems*, 36, 80744–80772.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019), 'Bert: Pre-training of deep bidirectional transformers for language understanding', in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Domini, G. (2020), 'Exhibitions, patents, and innovation in the early twentieth century: evidence from the Turin 1911 International Exhibition', *European Review of Economic History*, 24(3), 578–600.

Donges, A. & Selgert, F. (2019), 'Do legal differences matter? A comparison of German patent law regimes before 1877', *Jahrbuch für Wirtschaftsgeschichte/Economic History Yearbook*, 60(1), 57–92.

Donges, A. & Streb, J. (2024), 'Causes of German Inventiveness, 1815–1990. What We Can Learn from Patent Statistics', *German Economic Review*, 25(4), 301–323.

Dosovitskiy, A. (2020), 'An image is worth 16x16 words: Transformers for image recognition at scale', *arXiv preprint arXiv:2010.11929*.

Galor, O. (2011), *Unified Growth Theory*, Princeton University Press.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016), *Deep Learning*, Vol. 1, MIT Press Cambridge.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F. et al. (2025), 'Towards an AI co-scientist', *arXiv preprint arXiv:2502.18864*.

Greif, G., Griesshaber, N. & Greif, R. (2025), 'Multimodal LLMs for OCR, OCR Post-Correction, and Named Entity Recognition in Historical Documents', *arXiv preprint arXiv:2504.00414*.

Griesshaber, N. & Ogilvie, S. (2025), Transplanting Craft Guilds to Colonial Latin America: A Large Language Model Analysis, CEPR Discussion Paper 20556, CEPR, Paris and London.

Griliches, Z. (1990), 'Patent Statistics as Economic Indicators: A Survey', *Journal of Economic Literature*, 28(4), 1661–1707.

Guinnane, T., Harris, R., Lamoreaux, N. R. & Rosenthal, J.-L. (2007), 'Putting the Corporation in its Place', *Enterprise & Society*, 8(3), 687–729.

Gundlach, H., Lynch, J., Mertens, M. & Thompson, N. (2025), 'The Price of Progress: Algorithmic Efficiency and the Falling Cost of AI Inference', *arXiv preprint arXiv:2511.23455*.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A. et al. (2022), 'Training compute-optimal large language models', *arXiv preprint arXiv:2203.15556*.

Humphries, M., Leddy, L. C., Downton, Q., Legace, M., McConnell, J., Murray, I. & Spence, E. (2025), 'Unlocking the archives: Using large language models to transcribe handwritten historical documents', *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 1–19.

Jayes, J. (2025), *Like moths to a flame: an individual level approach to technological change in 20th century Sweden*, Doctoral Thesis (compilation), Lund University School of Economics and Management, Lund University, Lund.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A. et al. (2020), 'Scaling laws for neural language models', *arXiv preprint arXiv:2001.08361*.

Kim, J., Kang, S., Park, J., Kim, J. & Hwang, S. J. (2025), 'Interpreting Attention Heads for Image-to-Text Information Flow in Large Vision-Language Models', *arXiv preprint arXiv:2509.17588*.

Korinek, A. (2025), AI Agents for Economic Research, Working Paper 34202, National Bureau of Economic Research.

Lagakos, D., Michalopoulos, S. & Voth, H.-J. (2025), American Life Histories, Technical report, National Bureau of Economic Research.

LeCun, Y., Bengio, Y. & Hinton, G. (2015), 'Deep learning', *nature*, 521(7553), 436–444.

Lehmann-Hasemeyer, S. & Opitz, A. (2024), 'Data sources on the 19th and early 20th century German capital market: challenges and opportunities', *German Economic Review*, 25(4), 371–391.

Lehmann-Hasemeyer, S. & Streb, J. (2018), 'Does social security crowd out private savings? the case of bismarck's system of social insurance', *European Review of Economic History*, 22(3), 298–321.

Levchenko, M. (2025), 'Evaluating LLMs for Historical Document OCR: A Methodological Framework for Digital Humanities', *arXiv preprint arXiv:2510.06743*.

Long, J. & Ferrie, J. (2013), 'Intergenerational occupational mobility in Great Britain and the United States since 1850', *American Economic Review*, 103(4), 1109–1137.

Luo, W. (2025), *Multimodal-LLM as A Reliable Tool for Information Extraction from Historical Documents: A Digital Humanities Approach to Swedish Patent Cards (1945-1975)*, Master's thesis, Uppsala University, Disciplinary Domain of Humanities and Social Sciences, Faculty of Arts, Department of ALM.

Maddison, A. (2006), *Development Centre Studies The World Economy Volume 1: A Millennial Perspective and Volume 2: Historical Statistics*, Development Centre Studies, OECD Publishing.

Mitchener, L., Yiu, A., Chang, B., Bourdenx, M., Nadolski, T., Sulovari, A., Landsness, E. C., Barabasi, D. L. et al. (2025), 'Kosmos: An AI Scientist for Autonomous Discovery', *arXiv preprint arXiv:2511.02824*.

Mokyr, J. (2011), 'The gifts of Athena: Historical origins of the knowledge economy', in *The gifts of Athena*, Princeton University Press.

Moser, P. (2005), 'How do patent laws influence innovation? Evidence from nineteenth-century world's fairs', *American Economic Review*, 95(4), 1214–1236.

Moser, P. (2011), 'Do Patents Weaken the Localization of Innovations? Evidence from World's Fairs', *The Journal of Economic History*, 71(2), 363–382.

Moser, P. (2012), 'Innovation without Patents: Evidence from World's Fairs', *The Journal of Law and Economics*, 55(1), 43–74.

Moulton, D. & Severen, C. (2025), 'Harvesting Historical Data with LLMs', *Economic Insights*, 10(4), 1–6.

Rodrigues, E., Khalid, M., Nandi, S., Vrieze, A. & Yu, T. (2025), 'Benchmarking Methods for Digitizing Print Bibliographies', *Anthology of Computers and the Humanities*, 3, 1360–1371.

Ruggles, S. (2014), 'Big microdata for population research', *Demography*, 51(1), 287–297.

Shen, Z., Zhang, K. & Dell, M. (2020), 'A large dataset of historical japanese documents with complex layouts', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 548–549.

Sockin, J., Ash, E. & Shukla, S. (2025), Interviews, CESifo Working Paper 12229, CESifo.

Streb, J. (2023), 'Patent law and economic performance', *Rivista di storia economica*, 39(1), 3–26.

Streb, J. (2024), 'The Cliometric Study of Innovations', in *Handbook of Cliometrics*, Springer, 2225–2245.

Streb, J., Baten, J. & Yin, S. (2006), 'Technological and geographical knowledge spillover in the German empire 1877–1918', *The Economic History Review*, 59(2), 347–373.

Vafaie, M., Hertling, S., Banse-Strobel, I., Dubout, K. & Sack, H. (2025), 'End-to-end Information Extraction from Archival Records with Multimodal Large Language Models', in *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 6075–6083.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M. et al. (2022), 'Emergent abilities of large language models', *arXiv preprint arXiv:2206.07682*.

Wu, T.-H., Biamby, G., Quenum, J., Gupta, R., Gonzalez, J. E., Darrell, T. & Chan, D. M. (2025), 'Visual Haystacks: A Vision-Centric Needle-In-A-Haystack Benchmark', *arXiv preprint arXiv:2407.13766*.

Xie, Y., La Mela, M. & Tell, F. (2025), 'Multimodal LLM-assisted Information Extraction from Historical Documents: The Case of Swedish Patent Cards (1945-1975) and ChatGPT', in *The 9th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2025), March 5–7, 2025, Tartu, Estonia*, University of Oslo Library, 1–15.

Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E. et al. (2025), 'Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models', *Computational Linguistics*, 1–46.

Appendix A: Prompts

FIGURE A.1
PATENT ENTRY EXTRACTION PROMPT

```
You are a specialist data extraction AI. Your task is to process a single page from a German Imperial Patent Office
     register and extract the specified contents into a perfectly structured, flat JSON array. You must act with the
     precision of a world-class quantitative historian, ensuring absolute accuracy and adherence to the following rules.
     **HISTORICAL ACCURACY IS PARAMOUNT - extract everything exactly as written, preserving all abbreviations, spellings,
     and punctuation exactly as they appear in the original text.**

## I. Core Principles & Scope

### Extraction Area
You will **ONLY** extract information from the two main content columns on the page.

### IGNORE Absolutely:
*   **Running Page Headers:** The text at the very top of the page (e.g., "Kl. 18. Nr. 55711." or "Kl. 19. Nr. 57 185." at
     the top-left and top-right).
*   **Page Numbers:** Any numbers located in the center of the running header.
*   **Footer Information:** Any text at the bottom of the page that begins with an asterisk (*), is a correction
     ("Berichtigung"), or contains standalone numbers (e.g., "2*", "1917") that are clearly not part of the patent entries
     above.

### Reading & Processing Order
Your output must strictly follow this sequence:
1.  First, process the **LEFT** content column from top to bottom.
2.  Second, process the **RIGHT** content column from top to bottom.

The final JSON array must reflect this sequential order of extracted items.

## II. Extraction Rules: Identifying & Formatting Content

For each block of text you encounter in the two main content columns, you must classify it as one of the following and
     format it accordingly.

### A. Category Headings

*   **Identification:** A category heading is a line that signals a new patent class or subclass.
    *   A main class heading always starts with "Klasse", followed by a number and a title (e.g., "Klasse 19. Eisenbahn-
     und Brückenbau.").
    *   A subclass heading is typically just the class code itself, often on its own line (e.g., "15a." or "45i.").
*   **Extraction Rule:** From the identified heading, you must extract **ONLY** the class number/code and nothing else.
    *   If the heading is "Klasse 19. Eisenbahn- und Brückenbau", you will extract "19".
    *   If the heading is "18 b.", you will extract "18b".
    *   Clean the extracted code by removing any trailing periods or whitespace.
*   **JSON Output:** The extracted class code must be placed in an object with the key "category".
    > **Example:** {"category": "3"}
    > **Example:** {"category": "3a"}

### B. Patent Entries

*   **Identification of a Standard Entry:** A standard, complete patent entry is a paragraph that:
    *   Begins with a patent ID number (e.g., 55711.).
    *   The ID may be prefixed with an enumeration (e.g., 1.) or a dagger symbol in parentheses (e.g., (†)).
    *   Contains descriptive text about the patent holder, location, title, description, etc.
    *   Typically ends with a date and a code (e.g., 26. Februar 1890. A -- 252.).
    *   **Very rarely, additional information may appear after the date/code (e.g., "Priorität aus der Anmeldung in Ö
     sterreich vom 23/5 08 anerkannt.") - include this if present.**

*   **Extraction Rule for ALL Entries (Standard and Truncated):**
    *   Capture the text of the entry verbatim, preserving all original characters, spelling, and punctuation.
    *   **Line Break Handling:**
        *   Join words that are hyphenated across a newline (e.g., Hindersin-\nstrasse becomes Hindersinstrasse).
        *   Replace all other newlines within the entry's paragraph with a single space to form one continuous line of text.

*   **CRITICAL EXCEPTION: Truncated Entries**
    The single-page context means you will encounter incomplete entries. You **MUST** extract these as they appear.
```

```
    *   **Entry Truncated at the Top-Left:** If a column begins mid-paragraph (continuing from a previous page), extract
     that partial entry as-is.
    *   **Entry Truncated at the Bottom-Right:** If an entry is cut off at the bottom of a column (continuing to the next
     page), extract that partial entry as-is.
    *   **Entry Spanning Columns on the SAME Page:** If an entry starts at the bottom of the left column and its text
     continues at the top of the right column, you must create **TWO SEPARATE** entries in the JSON array. The first object
     will contain the part from the left column, and the second object will contain the part from the right column.

*   **JSON Output:** The formatted text of the entry must be placed in an object with the key "entry".
    > **Example:** {"entry": "55711. COOMES, M, F., Arzt, und A. W. HYDE, in Louisville, 6281/2 Fourth Street bezw. 828
     Second Street, Grafschaft Jefferson, Staat Kentucky, V. St. A.; Vertreter: C. PIEPER in Berlin N.W., Hindersinstrasse
     3. Verfahren zur Bereitung von Stahl. 26. Februar 1890. A -- 252."}

### C. Content to Explicitly IGNORE
*   **Lists of Patent Numbers:** After a category heading, you will often see a block of text that is just a list of patent
     numbers and codes (e.g., 54571 A 3. -- 54814 A 77. -- 54881 A 37. -- 54977 A 54.). **DO NOT** extract these lists that
     are just consisting of patent numbers and patent codes. These invalid patent entries only occur directly below
     category headings. **These lists can span multiple lines and form large blocks of text - ignore the entire block.**
     Dates in a new line below a valid patent entry belong to the valid patent entry.
*   **Reference Information ("Vgl."):** Any text that begins with "Vgl." (e.g., "Vgl. Kl. 12 P. R. 21188. - Kl. 45 P. R.
     17470, 20344.") that appears between patent entries and class headings. **DO NOT** extract this cross-reference
     information.

## III. JSON Output Specification (Strict)

Your entire output must be a single, syntactically perfect JSON array. Any deviation will result in failure.

*   **Array Structure:** The output must begin with [ and end with ].
*   **Object Structure:** Each element in the array is an object {}.
*   **Key-Value Pairs:** Each object must contain exactly **one** key-value pair. The key must be either "category" or
     "entry".
*   **Quotation:** **ALL** keys **MUST** be enclosed in double quotes (e.g., "entry"). All string values must be enclosed
     in double quotes.
*   **Delimiters:** Objects must be separated by a comma ,. The last object in the array must **NOT** have a trailing comma.
*   **Escaping:** Within a string value, any literal double quote " must be escaped as \".

---

**Example of Perfect Output Structure:**

[
  {
    "category": "18"
  },
  {
    "entry": "55711. COOMES, M, F., Arzt, und A. W. HYDE in Louisville, 6281/2 Fourth Street bezw. 828 Second Street,
     Grafschaft Jefferson, Staat Kentucky, V. St. A.; Vertreter: C. PIEPER in Berlin N.W., Hindersinstrasse 3. Verfahren
     zur Bereitung von Stahl. 26. Februar 1890. A -- 252."
  },
  {
    "entry": "56181. VERSEN, B., in Dortmund, Heiligerweg 8. Verfahren und Vorrichtung zur Herstellung von
     Bessemer-Birnen-Böden. (Zusatz zum Patente Nr. 30634.) 15. Mai 1890. A -- 363."
  },
  {
    "entry": "56195. ADAMS, CH., in Pittsburgh, 110 Diamond Street, Pennsylvania, V. St. A.; Vertreter: F. EDMUND THODE &
     KNOOP in Dresden, Amalienstrasse 5. Unmittelbare Darstellung von Eisen aus seinen Erzen. 9. Juli 1890. A -- 461."
  },
  {
    "entry": "56205. -- HERBERTZ, F. A., in Köln a. Rh. Schmelzofen mit Dampfstrahl. 12. August 1890. A -- 461."
  },
  {
    "category": "3a"
  },
  {
    "entry": "(†) 55539. -- FÜRSTENHEIM, C., in Berlin C., Jerusalemerstrasse 15. Zusammenklappbares Büstengestell. 5. Juni
     1890. A -- 249."
  }
]
```

```
**Final Instruction:**  Adhere to all rules. Your output must be **ONLY the JSON array** and nothing else. Before
     finishing, double-check your output to ensure it is perfectly valid JSON according to the specifications above.
```

Notes: The prompt was sent together with an image of our image corpus to Gemini-2.5-Pro in order to extract the patent entries and technological classes. Gemini-2.5-Pro returned a JSON object with keys *entry* and *category* in sequential reading order.

FIGURE A.2
REPARATION PROMPT

```
You are a data validator. Your task is to classify a German patent entry as either valid (‘1‘) or truncated (‘0‘).

**CRITICAL DISTINCTION:**
- A **valid** entry contains the END of the patent with date/registration information, regardless of whether it starts from
     the beginning or middle of the original patent text
- A **truncated** entry starts normally with a patent ID but is cut off mid-sentence and MISSES the end/date

**EXAMPLES:**

**TRUNCATED (tag as ‘0‘)** - These start normally but are cut off mid-sentence:
- "(†) 15322. -- SCHWINTZER & GRÄFF in Berlin S., Sebastianstr. 18. Neuerungen an Hänge- und Steh-Schiebelampen, bestehend
     in einem gläsernen, mit leicht zerlegbarer Metallumkleidung"
- "240938. Sulzer, Robert, Winterthur, Schweiz; Vertr.:"

**VALID (tag as ‘1‘)** - These contain the end with date/registration, even if they start mid-sentence:
- "Neuerungen in der Herstellung der sog. Rahmen für Schuhe und Stiefel. 28. Januar 1882. -- A 648."
- "der Ränder des Laufmantels von Luftradreifen. 17/8 13. -- A 1492."
- "zigerstr. 91. Zweibehälter-Drahtziehmaschine. 29/1 96."
- "mit auswechselbaren, ineinandergreifenden Segmenten mit seitlichen Hohlräumen und einem mittleren, über den seitlichen
     Hohlräumen angeordneten, unterstützten Hilfshohlraum. 2/3 13. -- A 2069. Priorität aus der Anmeldung in den
     Vereinigten Staaten von Amerika vom 27/7 12 anerkannt."

Respond with a single character only:
- ‘1‘ if the entry contains the end with date/registration information
- ‘0‘ if the entry is truncated mid-sentence and missing some part at the end

Do not provide any explanation or additional text.
```

Notes: This prompt was used to identify entries that were truncated after Gemini-2.5-Pro extracted the patent entries from the image scans. An extracted patent entry was appended to the bottom of this prompt and then sent to Gemini-2.5-Flash-Lite. This was done independently for all patent entries. The models returns “valid” (1) for any entry containing the closing date and registration code, even if the beginning is missing, while the model returns “truncated” (0) for entries that start properly but are cut off mid-sentence. Afterward, our pipeline merged truncated entries with the respective entry below.

FIGURE A.3
VARIABLE EXTRACTION PROMPT

```
You will be provided with a single text entry from Germany’s Imperial Patent Office. Your task is to function as a highly
     precise information extraction engine. You must carefully analyze the entry and extract specific fields, formatting
     the output as a single, valid JSON object.

**Core Principles:**
*   **Exactness:** All extracted text must be an *exact, character-for-character copy* from the source text. Do not add,
     remove, or alter any characters, including historical German characters (e.g., s).
*   **Structure:** The patent entries almost always follow a specific order: ‘patent_id‘ -> ‘name(s)‘ -> ‘location(s)‘ ->
     (optional ‘Vertreter‘ info) -> ‘description‘ -> date -> (optional addendum note or priority registration). Use this
     predictable structure to guide your extraction.
*   **Output Format:** The output MUST be a single JSON object with the keys: ‘patent_id‘, ‘name‘, ‘location‘,
     ‘description‘, and ‘date‘. If a value cannot be reliably found for a field, use the string "NaN".

---

**Detailed Extraction Rules:**

**1. ‘patent_id‘ (String)**
```

```
*   **Position:** The patent ID is the numerical identifier at the beginning of the entry.
*   **Format:** It is a one- to six-digit number.
*   **Rule:** You MUST ignore any non-digit prefixes. This includes list enumerations (e.g., ‘1.‘, ‘17.‘), symbols (e.g.,
     ‘(†)‘), dashes, or spaces. Extract ONLY the numeric digits.
*   **Example:** For ‘(†) 2. 35321. -- ...‘, you must extract ‘"35321"‘.

**2. ‘name‘ (String)**
*   **Position:** This is the name of the patent holder(s), which immediately follows the ‘patent_id‘.
*   **Rule:** Extract the full name(s) exactly as written, but follow these CRITICAL rules:
    *   **INCLUDE:** Corporate forms (e.g., ‘AKTIEN-GESELLSCHAFT FÜR BERGBAU- UND ZINKHÜTTENBETRIEB‘).
    *   **INCLUDE:** All parts of compound names (e.g., ‘SCHIFFER & KIRCHER‘).
    *   **INCLUDE:** Academic titles (e.g., ‘Dr.‘, ‘Dr.-Ing.‘, ‘Ing.‘) as they are part of the person’s formal name.
    *   **INCLUDE:** The word "Firma" if it appears as part of the company name (e.g., ‘Firma C. KESSELER‘).
    *   **CRITICAL - EXCLUDE OCCUPATIONS:** NEVER include occupational descriptions or job titles, such as:
        *   ‘Hofschlächter‘, ‘Professor‘, ‘Ingenieur‘, ‘Kaufmann‘, ‘Fabrikant‘, ‘Mechaniker‘
        *   ‘Nachfolger‘, ‘Marine-Lieutenant‘, or any other job titles
    *   **CRITICAL - EXCLUDE LOCATIONS:** NEVER include location information in the name field. For multiple patent
     holders, extract ONLY the names and separate them with ‘und‘ or ‘,‘ as they appear in the original text.
*   **CRITICAL:** This field is for the patent holder ONLY. Do NOT extract the name of the representative (‘Vertreter‘).
     The representative’s name, if present, appears later in the text and must be ignored for this field.

**3. ‘location‘ (String)**
*   **Position:** This is the location of the patent holder(s), and it almost always follows the holder’s ‘name‘ directly.
     It is often introduced by a preposition like ‘in‘ or ‘auf‘.
*   **Rule:** Extract the entire location phrase associated with the patent holder(s).
    *   Include all details: city, state, country (e.g., ‘Kladno, Böhmen‘, ‘Chicago, V. St. A.‘).
    *   **CRITICAL - MULTIPLE PATENT HOLDERS:** When there are multiple patent holders, each with their own location,
     concatenate ALL locations into a single string, separated by ‘und‘ or ‘,‘ as they appear in the original text.
    *   **Example:** For ‘RUHM, H., in Sulko Zechc bei Pilsen und L. WOLF in Prag-Karolinenthal‘, the location should be
     ‘Sulko Zechc bei Pilsen und Prag-Karolinenthal‘.
*   **CRITICAL:** Do NOT extract the location of the representative (‘Vertreter‘). The correct location is the one directly
     linked to the ‘name‘ you extracted in the previous step.

**4. ‘description‘ (String)**
*   **Position:** This is the patent’s subject matter. It is the text located *after* the holder’s name/location and
     *before* the final date.
*   **Rule:** Extract the core descriptive text. This can be as short as a single word. You must apply the following two
     exclusion rules:
    *   **Exclusion 1: Representative Information:** You MUST completely exclude any block of text detailing a
     representative. This block often starts with ‘Vertreter:‘ and includes their name, title, and location.
    *   **Exclusion 2: Addendum Notes:** You MUST completely exclude any parenthetical notes about patent addendums after
     or at the end of the patent description. These notes look like ‘(Zusatz zum Patent 12345.)‘ or ‘(II. Zusatz zum Patent
     17502.)‘ and are usually at the end or after the patent description.
*   After applying these exclusions, extract the remaining text for the description.

**5. ‘date‘ (String)**
*   **Position:** The date is located near the end of the entry.
*   **Rule:** Extract the date string *exactly* as it appears. If multiple dates are present in the entry, the correct one
     is always the date referring to the German registration.

---

**Example Input Entry:**
35 321. -- KARLIK, JOH., in Kladno, Böhmen; Vertreter: C. FEHLERT & G. LOUBIER, Firma: C. KESSELER in Berlin SW.,
     Anhaltstr. 6. Wipper mit verschiedenen, sich während einer Umdrehung ändernden Umfangsgeschwindigkeiten. 5. November
     1885.

**Correct JSON Output:**
{
  "patent_id": "35321",
  "name": "KARLIK, JOH.",
  "location": "Kladno, Böhmen",
  "description": "Wipper mit verschiedenen, sich während einer Umdrehung ändernden Umfangsgeschwindigkeiten.",
  "date": "5. November 1885"
}

**Second Example Input Entry:**
15948. -- ,,VIEILLE MONTAGNE" AKTIEN-GESELLSCHAFT FÜR BERGBAU- UND ZINKHÜTTENBETRIEB in Altenberge. Neuerungen an
     ringförmigen Setzmaschinen (Zusatz zu P. R. 15224). 25. März 1881. -- A 789.
```

```
**Correct JSON Output:**
{
  "patent_id": "15948",
  "name": "\"VIEILLE MONTAGNE\" AKTIEN-GESELLSCHAFT FÜR BERGBAU- UND ZINKHÜTTENBETRIEB",
  "location": "Altenberge",
  "description": "Neuerungen an ringförmigen Setzmaschinen",
  "date": "25. März 1881"
}

**Third Example Input Entry (Multiple Patent Holders and Locations):**
15203. -- RUHM, H., in Sulko Zechc bei Pilsen und L. WOLF in Prag-Karolinenthal; Vertreter: C. KESSELER in Berlin W.,
     Mohrenstr. 63 I. Endloser Planherd. 1. April 1881. -- A 609.

**Correct JSON Output:**
{
  "patent_id": "15203",
  "name": "RUHM, H. und L. WOLF",
  "location": "Sulko Zechc bei Pilsen und Prag-Karolinenthal",
  "description": "Endloser Planherd.",
  "date": "1. April 1881"
}

**The entry from which you should extract information:**
```

Notes: This prompt was used to extract the five desired variables. A repaired patent entry was appended to the bottom of this prompt and then sent to Gemini-2.5-Flash-Lite. The LLM returned a JSON object with keys *patent_id*, *name*, *location*, *description*, and *date*. The variables were then appended to the respective patent entry in the dataset. This was done indepdently for all patent entries.

Appendix B: Prompts for Special Volumes with Different Layout

FIGURE B.1 SPECIAL VOLUMES WITH DIFFERENT LAYOUT: PATENT ENTRY EXTRACTION PROMPT

```
You are a specialist data extraction AI. Your task is to process a single page from a German Imperial Patent Office
     register and extract the specified contents into a perfectly structured, flat JSON array. You must act with the
     precision of a world-class quantitative historian, ensuring absolute accuracy and adherence to the following rules.
     **HISTORICAL ACCURACY IS PARAMOUNT - extract everything exactly as written, preserving all abbreviations, spellings,
     and punctuation exactly as they appear in the original text.**

## I. Core Principles & Scope

### Extraction Area
You will **ONLY** extract information from the two main content columns on the page.

### IGNORE Absolutely:
*   **Running Page Headers:** The text at the very top of the page (e.g., "Kl. 18. Nr. 55711." or "Kl. 19. Nr. 57 185." at
     the top-left and top-right).
*   **Page Numbers:** Any numbers located in the center of the running header.
*   **Footer Information:** Any text at the bottom of the page that begins with an asterisk (*), is a correction
     ("Berichtigung"), or contains standalone numbers (e.g., "2*", "1917") that are clearly not part of the patent entries
     above.

### Reading & Processing Order
Your output must strictly follow this sequence:
1.  First, process the **LEFT** content column from top to bottom.
2.  Second, process the **RIGHT** content column from top to bottom.

The final JSON array must reflect this sequential order of extracted items.

## II. Extraction Rules: Identifying & Formatting Content

For each block of text you encounter in the two main content columns, you must classify it as one of the following and
     format it accordingly.

### A. Category Headings

*   **Identification:** A category heading is a line that signals a new patent class.
    *   A main class heading always starts with "Klasse", followed by a number and a title (e.g., "Klasse 19. Eisenbahn-
     und Brückenbau.").
*   **Extraction Rule:** From the identified heading, you must extract **ONLY** the class number and nothing else.
    *   If the heading is "Klasse 19. Eisenbahn- und Brückenbau", you will extract "19".
    *   Clean the extracted code by removing any trailing periods or whitespace.
*   **JSON Output:** The extracted class code must be placed in an object with the key "category".
    > **Example:** {"category": "3"}
    > **Example:** {"category": "19"}

### B. Patent Entries

*   **Identification of a Standard Entry:** A standard, complete patent entry is a paragraph that:
    *   Contains descriptive text about the patent holder, location, title, description, etc.
    *   Always ends with a patent registration number in the format "P. R. XXXX." (where XXXX is a number between 1 and
     9999, e.g., "P. R. 6201.").

*   **Extraction Rule for ALL Entries (Standard and Truncated):**
    *   Capture the text of the entry verbatim, preserving all original characters, spelling, and punctuation.
    *   **Line Break Handling:**
        *   Join words that are hyphenated across a newline (e.g., Hindersin-\nstrasse becomes Hindersinstrasse).
        *   Replace all other newlines within the entry's paragraph with a single space to form one continuous line of text.

*   **CRITICAL EXCEPTION: Truncated Entries**
    The single-page context means you will encounter incomplete entries. You **MUST** extract these as they appear.
    *   **Entry Truncated at the Top-Left:** If a column begins mid-paragraph (continuing from a previous page), extract
     that partial entry as-is.
    *   **Entry Truncated at the Bottom-Right:** If an entry is cut off at the bottom of a column (continuing to the next
     page), extract that partial entry as-is.
    *   **Entry Spanning Columns on the SAME Page:** If an entry starts at the bottom of the left column and its text
     continues at the top of the right column, you must create **TWO SEPARATE** entries in the JSON array. The first object
     will contain the part from the left column, and the second object will contain the part from the right column.
```

```
*   **JSON Output:** The formatted text of the entry must be placed in an object with the key "entry".
    > **Example:** {"entry": "COOMES, M, F., Arzt, und A. W. HYDE, in Louisville, 6281/2 Fourth Street bezw. 828 Second
     Street, Grafschaft Jefferson, Staat Kentucky, V. St. A.; Vertreter: C. PIEPER in Berlin N.W., Hindersinstrasse 3.
     Verfahren zur Bereitung von Stahl. 26. Februar 1890. P. R. 6201."}

### C. Content to Explicitly IGNORE
*   **Lists of Patent Numbers:** After a category heading, you will often see a block of text that is just a list of patent
     numbers and codes (e.g., 54571 A 3. -- 54814 A 77. -- 54881 A 37. -- 54977 A 54.). **DO NOT** extract these lists that
     are just consisting of patent numbers and patent codes. These invalid patent entries only occur directly below
     category headings. **These lists can span multiple lines and form large blocks of text - ignore the entire block.**
     Dates in a new line below a valid patent entry belong to the valid patent entry.
*   **Reference Information ("Vgl."):** Any text that begins with "Vgl." (e.g., "Vgl. Kl. 12 P. R. 21188. - Kl. 45 P. R.
     17470, 20344.") that appears between patent entries and class headings. **DO NOT** extract this cross-reference
     information.

## III. JSON Output Specification (Strict)

Your entire output must be a single, syntactically perfect JSON array. Any deviation will result in failure.

*   **Array Structure:** The output must begin with [ and end with ].
*   **Object Structure:** Each element in the array is an object {}.
*   **Key-Value Pairs:** Each object must contain exactly **one** key-value pair. The key must be either "category" or
     "entry".
*   **Quotation:** **ALL** keys **MUST** be enclosed in double quotes (e.g., "entry"). All string values must be enclosed
     in double quotes.
*   **Delimiters:** Objects must be separated by a comma ,. The last object in the array must **NOT** have a trailing comma.
*   **Escaping:** Within a string value, any literal double quote " must be escaped as \".

---

**Example of Perfect Output Structure:**

[
  {
    "category": "18"
  },
  {
    "entry": "COOMES, M, F., Arzt, und A. W. HYDE in Louisville, 6281/2 Fourth Street bezw. 828 Second Street, Grafschaft
     Jefferson, Staat Kentucky, V. St. A.; Vertreter: C. PIEPER in Berlin N.W., Hindersinstrasse 3. Verfahren zur Bereitung
     von Stahl. 26. Februar 1890. P. R. 6201."
  },
  {
    "entry": "VERSEN, B., in Dortmund, Heiligerweg 8. Verfahren und Vorrichtung zur Herstellung von Bessemer-Birnen-Böden.
     (Zusatz zum Patente Nr. 30634.) 15. Mai 1890. P. R. 6202."
  },
  {
    "entry": "ADAMS, CH., in Pittsburgh, 110 Diamond Street, Pennsylvania, V. St. A.; Vertreter: F. EDMUND THODE & KNOOP in
     Dresden, Amalienstrasse 5. Unmittelbare Darstellung von Eisen aus seinen Erzen. 9. Juli 1890. P. R. 6203."
  },
  {
    "entry": "HERBERTZ, F. A., in Köln a. Rh. Schmelzofen mit Dampfstrahl. 12. August 1890. P. R. 6204."
  },
  {
    "category": "19"
  },
  {
    "entry": "FÜRSTENHEIM, C., in Berlin C., Jerusalemerstrasse 15. Zusammenklappbares Büstengestell. 5. Juni 1890. P. R.
     6205."
  }
]

**Final Instruction:**  Adhere to all rules. Your output must be **ONLY the JSON array** and nothing else. Before
     finishing, double-check your output to ensure it is perfectly valid JSON according to the specifications above.
```

Notes: The prompt used to extract patent entries and technological classes from images in volumes 1877–8 and 1879, which have a different layout from the other volumes. Please see the notes in Figure A.1 for more information.

FIGURE B.2 SPECIAL VOLUMES WITH DIFFERENT LAYOUT: REPARATION PROMPT

```
You are a data validator. Your task is to classify a German patent entry as either complete (‘1‘) or truncated (‘0‘).

A **complete (valid)** entry must end with a date followed by a patent registration number. Due to OCR errors, the
     registration format may appear in many variations including "P. R. XXXX.", "I. R. XXXX.", "R. XXXX.", "R. R. XXXX.",
     "P. R. Nr. XXXX", "P. R. Nr. XXXX.", etc. (where XXXX is a number between 1 and 9999). The key is that it ends with a
     date and some form of registration number. An entry is complete even if it’s truncated at the beginning, as long as it
     ends properly.

**Examples of COMPLETE entries (should be marked as ‘1‘):**
- ‘HAHLWEG, C. Werkzeug zur Herstellung von Steinfassungen für Taschenuhren. 22. Juli 1877. P. R. 80.‘
- ‘LÖWIG, G., & LÖWIG, Fr. Verfahren zur Darstellung von Aetzalkalien und Thonerdepräparaten. 3. Juli 1877. P. R. 93.‘
- ‘MORGAN RICHARDS, J. Herstellung durchbohrter Pillen und der zu ihrer Anfertigung nöthigen Maschine. 3. August 1877. P.
     R. 134.‘
- ‘anilin und anderen tertiären aromatischen Monaminen. 15. Dezember 1877. P. R. 1886.‘ (truncated at beginning but
     complete)
- ‘SCHRÖDER, C. Windmühlenflügel mit durch Federn geöffneten, durch den Winddruck geschlossenen Klappen. 13. Januar 1878.
     I. R. 1843.‘ (OCR error: I. R. instead of P. R.)
- ‘BERNHARDI SOHN, Dr., DRAENERT, G. E. JACKSON’sche Wendevorrichtung an horizontalen Windrädern mit Kettenbetrieb. 13.
     November 1877. R. 1229.‘ (OCR error: R. instead of P. R.)
- ‘SCHNEIDER & JACQUED. Turbine mit eingesenkten Zwischenschaufeln. 4. September 1877. R. 547.‘ (OCR error: R. instead of
     P. R.)
- ‘FISCHER, G. A. Niederschraubhahn mit und ohne Entleerungsventil, mittelst dessen eine unter Druck stehende Dampf- oder
     Wasserleitung angebohrt werden kann. 20. Oktober 1877. R. R. 874.‘ (OCR error: R. R. instead of P. R.)
- ‘GROSSCHOPFF, DR. C. Selbstthätiger Desinfector für Aborte. 31. März 1878. P. R. Nr. 3197‘ (OCR variation: P. R. Nr.
     without final period)

An entry is **truncated (invalid)** if it is missing the date or patent registration number at the very end.

**Examples of TRUNCATED entries (should be marked as ‘0‘):**
- ‘BARDIN, G. Verfahren, die natürliche Feder zu verzwirnen und die Verwendung solcher Feder-Che‘
- ‘EHESTÄDT und ROBERT. Beweglicher Arm an Beleuchtungsapparaten aus in Form eines Parallelo-‘

**CRITICAL:** When in doubt, mark as ‘1‘ (complete). Only mark as ‘0‘ if the entry is clearly truncated and missing both
     date and registration number at the end.

Respond with a single character only:
- ‘1‘ if the entry is complete (ends with date and any form of registration number, regardless of OCR variations).
- ‘0‘ if the entry is truncated/invalid (missing date and registration number at the end).

Do not provide any explanation or additional text.
```

Notes: The prompt used to repair patent entries from volumes 1877–8 and 1879, which have a different layout from the other volumes. Please see the notes in Figure A.2 for more information.

FIGURE B.3 SPECIAL VOLUMES WITH DIFFERENT LAYOUT: VARIABLE EXTRACTION PROMPT

```
You will be provided with a single text entry from Germany’s Imperial Patent Office. Your task is to function as a highly
     precise information extraction engine. You must carefully analyze the entry and extract specific fields, formatting
     the output as a single, valid JSON object.

**Core Principles:**
*   **Exactness:** All extracted text must be an *exact, character-for-character copy* from the source text. Do not add,
     remove, or alter any characters, including historical German characters (e.g., s).
*   **Structure:** The patent entries almost always follow a specific order: ‘patent_id‘ -> ‘name(s)‘ -> ‘location(es)‘ ->
     (optional ‘Vertreter‘ info) -> ‘description‘ -> (optional addendum note) -> ‘date‘ -> (optional addendum note) -> ‘P.
     R. XXXX.‘ (registration number). Use this predictable structure to guide your extraction.
*   **Output Format:** The output MUST be a single JSON object with the keys: ‘patent_id‘, ‘name‘, ‘location‘,
     ‘description‘, and ‘date‘. If a value cannot be reliably found or identified for a field, use the string "NaN".

---

**Detailed Extraction Rules:**

**1. ‘patent_id‘ (String)**
```

```
*   **Location:** The patent ID is the numerical identifier at the end of the entry in the format "P. R. XXXX.".
*   **Format:** It is a number between 1 and 9999.
*   **Rule:** Extract ONLY the numeric digits from the "P. R. XXXX." format at the end of the entry.
*   **Example:** For `... 5. November 1885. P. R. 6201.`, you must extract `"6201"`.

**2. `name` (String)**
*   **Location:** This is the name of the patent holder(s), which appears at the beginning of the entry.
*   **Rule:** Extract the full name(s) exactly as written.
    *   Include corporate forms (e.g., `AKTIEN-GESELLSCHAFT FÜR BERGBAU- UND ZINKHÜTTENBETRIEB`).
    *   Include academic or professional titles (e.g., `Dr.-Ing.`).
    *   Include all parts of compound names (e.g., `SCHIFFER & KIRCHER`).
    *   **CRITICAL:** Do NOT extract occupations or job descriptions that may appear with names. Extract only the actual names and titles.
*   **CRITICAL:** This field is for the patent holder ONLY. Do NOT extract the name of the representative (`Vertreter`). The representative's name, if present, appears later in the text and must be ignored for this field.

**3. `location` (String)**
*   **Location:** This is the location of the patent holder, and it almost always follows the holder's `name` directly. It is often introduced by a preposition like `in` or `auf`.
*   **Rule:** Extract the entire location phrase associated with the patent holder(s).
    *   Include all details: city, state, country (e.g., `Kladno, Böhmen`, `Chicago, V. St. A.`).
*   **CRITICAL:** Do NOT extract the location of the representative (`Vertreter`). The correct location is the one directly linked to the `name` you extracted in the previous step.

**4. `description` (String)**
*   **Location:** This is the patent's subject matter. It is the text located *after* the holder's name/location and *before* the final date.
*   **Rule:** Extract the core descriptive text. This can be as short as a single word. You must apply the following two exclusion rules:
    *   **Exclusion 1: Representative Information:** You MUST completely exclude any block of text detailing a representative. This block often starts with `Vertreter:` and includes their name, title, and location.
    *   **Exclusion 2: Addendum Notes:** You MUST completely exclude any parenthetical notes about patent addendums. These notes look like `(Zusatz zum Patent 12345.)` or `(II. Zusatz zum Patent 17502.)` and are usually found at the end of the description, just before the date.
*   After applying these exclusions, extract the remaining text for the description.

**5. `date` (String)**
*   **Location:** The date is located near the end of the entry, before the final "P. R. XXXX." registration number.
*   **Rule:** Extract the date string *exactly* as it appears. If multiple dates are present in the entry, the correct one is always the date referring to the German registration.

---

**Example Input Entry:**
KARLIK, JOH., in Kladno, Böhmen; Vertreter: C. FEHLERT & G. LOUBIER, in Firma: C. KESSELER in Berlin SW., Anhaltstrafse 6.
     Wipper mit verschiedenen, sich während einer Umdrehung ändernden Umfangsgeschwindigkeiten. 5. November 1885. P. R.
     6201.

**Correct JSON Output:**
{
  "patent_id": "6201",
  "name": "KARLIK, JOH.",
  "location": "Kladno, Böhmen",
  "description": "Wipper mit verschiedenen, sich während einer Umdrehung ändernden Umfangsgeschwindigkeiten.",
  "date": "5. November 1885."
}

**The entry from which you should extract information:**
```

Notes: The prompt used to extract variables from patent entries stemming from volumes 1877–8 and 1879, which have a different layout from the other volumes. Please see the notes in Figure A.3 for more information.

Oxford Economic and Social History Working Papers

is edited by

Victoria Gierok,
Nuffield College, Oxford, OX1 1NF