Paper ML14

Large Language Models as Medical Writers: Challenges and Solutions

Ioannis Reklos, IQVIA, Athens, Greece
Alexandrina Lambova, IQVIA, Sofia, Bulgaria
Kiril Matev, IQVIA, Sofia, Bulgaria
Ketevan Rtveladze, IQVIA, London, United Kingdom
Julia Gallinaro, IQVIA, London, United Kingdom
Ines Guerra, IQVIA, London, United Kingdom

ABSTRACT

The creation of Systematic Literature Review (SLR) reports and Health Technology Assessment (HTA) dossiers is crucial in medical writing, often requiring extensive expert effort. We examined the use of Large Language Models (LLMs) to automate key steps, specifically information extraction and medical text generation. We identified core challenges in AI-assisted document creation. For information extraction, issues arose when information appeared in text and tables. We addressed table-related problems, such as multi-page splits and recognition errors, through custom solutions, achieving substantial improvements. Text-related challenges included topic-specific prompt tuning and Retrieval Augmented Generation (RAG) failures. We mitigated these with topic classifiers for improved retrieval. Extraction accuracy was assessed against human-generated gold standards. In text generation, we faced challenges with content synthesis and granularity control. We implemented a structured outline approach to guide style and organization. This paper outlines challenges, solutions, and future steps toward AI-assisted medical writing.

INTRODUCTION

Systematic Literature Reviews (SLRs) and Health Technology Assessment (HTA) dossiers are cornerstone artefacts in evidence generation, requiring hundreds of expert hours to produce. At the same time, they remain difficult to automate because their source material is highly heterogeneous—narrative text, semi-structured tables, and figures dispersed across long, scanned, or digitally born PDFs. Automating even part of this workflow demands models that can (i) reliably extract fine-grained facts from both text and tables, (ii) retrieve the correct evidence under strict traceability requirements, and (iii) synthesize regulator-ready prose with controllable scope and granularity. Large Language Models (LLMs) and retrieval-augmented generation (RAG) provide a compelling foundation, but naïve application in biomedical evidence synthesis exposes failure modes—table fragmentation, header or cell misrecognition, prompt brittleness, retrieval noise, and uncontrolled content expansion—that undermine trust and reproducibility.

This paper investigates the use of LLMs to automate two high-impact steps in SLR and HTA development: information extraction and medical text generation. We treat documents as multimodal, structure-rich objects rather than flat text and design a pipeline that explicitly identifies where errors occur and how they can be mitigated. For extraction, we address table-specific challenges such as multi-page splits, spanning cells, and recognition errors through custom stitching and normalization procedures, supported by structure-aware indexing that treats tables and text as first-class, linkable units. For text generation, we automate fact extraction and constrain drafting through a structured outline that encodes section intent, permissible evidence types, and citation granularity—reducing uncontrolled expansion and improving consistency with target dossier templates.

We evaluate the proposed framework across its three principal components—extraction, retrieval, and generation—to assess robustness and practical benefit. Specifically, we test whether structured table extraction enhances information recall, topic-aware retrieval improves evidence relevance, and outline-guided generation strengthens factual control and stylistic consistency. The results show consistent gains in accuracy, efficiency, and traceability, demonstrating that domain-specific, structure-aware integration of LLMs can materially advance automation in evidence synthesis while preserving auditability and alignment with regulatory expectations.

The scope of this work is deliberately pragmatic: we target the precision-critical portions of SLR and HTA authoring where automation can yield measurable efficiency gains while maintaining auditability. We do not replace expert authorship; instead, we propose an architecture that surfaces provenance at the document level and supports human-in-the-loop verification.

Our contributions are the following: a) Structure-aware extraction pipeline that reconciles text and tables, including multi-page table stitching and header/cell reconstruction, yielding substantial improvements over off-the-shelf parsers

for biomedical PDFs. b) Retrieval robustness via topic classifiers that improve upon RAG, reducing retrieval failures and improving evidence relevance for downstream extraction and drafting. c) Controllable fact extraction and generation using outline-guided prompts that enforce section scope, evidence requirements, citation behavior, and granularity, improving stylistic consistency for SLR/HTA narratives.

TABLE EXTRACTION

We developed an automated table extraction pipeline for scientific PDF documents that leverages large language models (LLMs) with vision and text capabilities. The methodology combines document parsing, image-based table recognition, text correction, and table merging into a unified framework.

1. DOCUMENT PARSING

The first step in the pipeline is to parse the PDF file input. Depending on the configuration, the parser can operate in text or image mode. In text mode, the full text of the document is extracted and stored per page as well as in an aggregated file. In image mode, each page is rendered as a high-resolution image (PNG format, 4× scaling matrix) in addition to text extraction. These intermediate outputs are stored in structured project directories for downstream processing.

2. IMAGE ENCODING AND LLM INTERACTION

To enable table recognition, each page image is converted into a base64-encoded string and passed to a GPT-4o model [1] through the Azure OpenAI API. A system prompt constrains the model to return results in strict JSON format with the following schema:

- title: table title, including table number when available
- columns: list of column headers
- rows: list of dictionaries representing row values keyed by column name
- footnotes: list of table footnotes

If no table is present on the page, the model is instructed to return a [NONE] token, thereby avoiding false positives.

3. ERROR CORRECTION VIA CROSS-MODAL VALIDATION

Since image-only extraction may produce transcription errors (e.g., numeric misreads or missing headers), we implemented a second LLM pass that integrates the page's markdown text representation. The initially extracted JSON is corrected against the textual content of the page, ensuring higher fidelity with the source document. This two-step process (vision extraction followed by text-based correction) mitigates OCR-like errors and improves table accuracy.

4. TABLE MERGING ACROSS PAGES

Many scientific publications split large tables across multiple pages. To reconstruct complete tables, we implemented an LLM-driven merging procedure. Extracted tables are compared pairwise for column similarity and row consistency. The merging prompt asks the LLM to return a unified table only when:

1. Column headers are highly similar.
2. Row values share consistent data types.
3. The continuation is explicitly supported by page content.

To avoid over-merging, similarity is further validated by calculating token-level matches between overlapping rows of the original and the merged tables. Only when similarity exceeds a defined threshold are the merged tables retained; otherwise, they individual tables are stored independently.

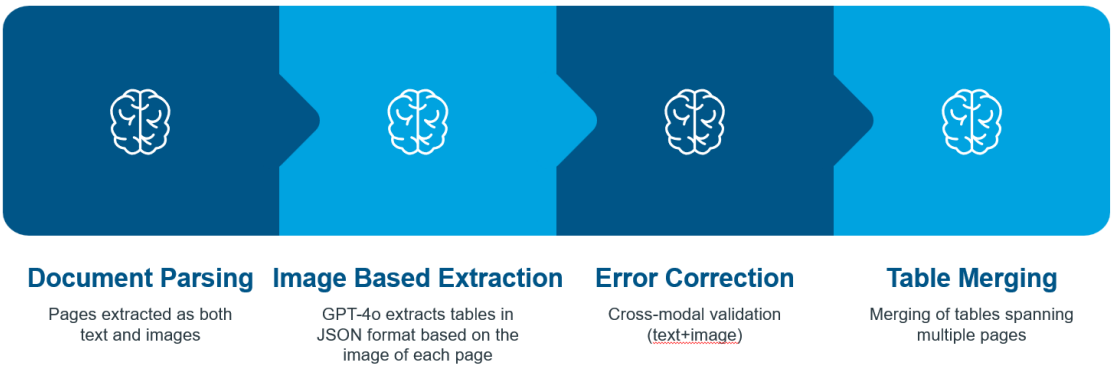

Figure 1: Overview of the Table Extraction pipeline

5. EVALUATION

We evaluated the proposed pipeline against extraction queries defined by subject matter experts (SMEs) to assess whether the integration of extracted tables improves retrieval and response accuracy. Two comparison scenarios were considered: (i) extraction prompt + PDF text only, and (ii) extraction prompt + PDF text + extracted tables. The evaluation set comprised 9 queries targeting economic model inputs and 32 queries targeting economic model outcomes, yielding a total of 41 SME-authored questions.

Each query was labelled by SMEs into one of four outcome categories: improved, worsened or unchanged. We then calculated the proportion of improvement as:

$$\%\ \boldsymbol{improvement} = \frac{\boldsymbol{\#\ improved\ -\ \#\ worsened}}{\boldsymbol{\#\ not\ working}} \times \mathbf{100}$$

The results are summarized in Table 1. For model inputs, 6 of 9 queries were not working under the baseline text-only condition. When table extraction was added, 2 queries improved, 1 query worsened, and 3 remained unchanged, resulting in a net improvement rate of 16.67%. For model outcomes, 9 of 32 queries failed under baseline conditions. With table integration, 7 queries improved, 2 worsened, and 23 remained unchanged, corresponding to a net improvement of 55.56%.

Topic	Total Questions	Not Working	Improved	Worsened	Unchanged	% Improvement
Model Inputs	9	6	2	1	3	16.67%
Model Outcomes	32	9	7	2	23	55.56%

Table 1: Evaluation of pipeline performance across SME-authored queries on model inputs (9 queries) and model outcomes (32 queries). Results compare baseline text-only extraction against text + extracted tables.

These results indicate that while gains on input-related queries were modest, outcome-related queries benefited substantially from the integration of structured tables, more than halving the proportion of non-working queries. This suggests that tables are especially valuable for capturing outcome information that is often fragmented or inconsistently represented in text alone. Importantly, the small number of worsened cases highlights the need for continued refinement of stitching and alignment procedures to minimize noise introduced by automatically reconstructed tables.

Overall, the evaluation demonstrates that combining extracted tables with text improves retrieval robustness and reduces the SME burden in answering evidence queries, with the strongest effect observed for clinical outcome data.

TOPIC CLASSIFIERS

DATA AND TASK FORMULATION

We study topic-level retrieval from clinical-style documents under severe data scarcity (~25 records). Each record contains three topics: Intervention, Patient characteristics and Objective. For each retained topic we use a pre-computed OpenAI text embedding $x \in R^d$

Let the label space be $\mathcal{Y} = \{\text{Intervention}, \text{Patient\_characteristics}, \text{Objective}\}$. From our training data we construct a dataset $\mathcal{D} = \{(x_j, y_j, g_j)\}_{j=1}^{n}$ where $x_j \in R^d$ is an embedding parsed from the stringified list, $y_j \in \mathcal{Y}$ is the topic label and $g_j \in \{1, \dots, G\}$ is a group identifier (the original document ID) used to prevent cross-document leakage in validation. The query dataset, which can be seen in Table 2, contains exactly one query embedding per class; we denote these by $\{q_c \in R^d : c \in \mathcal{Y}\}$. These queries are fixed across folds and represent class-specific information needs (e.g., "What is the intervention in the provided documents?").We address two problems: (i) multi-class classification of topic embeddings; (ii) class-aware retrieval for RAG given a class-specific query.

Topic	Query
Intervention	What is the intervention in the provided documents?
Patient Characteristics	What are the patient characteristics in the provided documents?
Objective	What is the objective of the studies?

Table 2: The query dataset containing one query per topic (and the corresponding embedding) to be used for cosine similarity-based retrieval.

PREPROCESSING

All learned systems are implemented as scikit-learn [2] Pipelines to eliminate leakage between training and test folds.

We first unit-normalize vectors to operate in cosine geometry:

$$\tilde{x} = \frac{x}{||x||_2}, \qquad \widetilde{q_c} = \frac{q_c}{||q_c||_2}$$

We then use a custom fold-safe PCA [3] module (SafePCA) with whitening. Given requested components k, SafePCA caps k at $\min(n_{\text{train}}, d)$ within each inner fold, preventing failures on small splits:$k_{\text{fit}} = \min\{k, n_{\text{train}}, d\}$.We tune $k \in \{8,16\}$. Whitening decorrelates features and can reduce variance in small-n regimes.

For the cosine baseline we use raw embedding space with only L2 normalization (no PCA), reflecting common practice in embedding retrieval.

MODELS AND HYPERPARAMETERS

We focus on simple, strongly regularized linear models that perform well in high-dimensional, small-n settings and support probabilities:

- Logistic Regression (LR): L_2-regularized [4], with balanced class weights.

- Linear SVM [5] with probability calibration (sigmoid, 3-fold) to obtain calibrated $p(y \mid x)$ [6].

Regularization strength is tuned over $C \in \{0.05, 0.1, 0.5, 1.0\}$.

PCA and classifier hyperparameters are tuned jointly inside the pipeline.

MODEL SELECTION AND VALIDATION PROTOCOL

We employ nested, group-aware cross-validation to obtain unbiased estimates and prevent leakage among topics from the same document:

- Outer loop: Group K-Fold with k=5 over g_j produces 5 disjoint test folds at document level.

- Inner loop: Group K-Fold k=4 on the outer loop train data drives GridSearchCV, refitting on macro-F1 score (treats classes evenly).

Since each document contributes multiple topic embeddings (e.g., *Intervention*, *Patient characteristics*, *Objective*), splitting at the level of individual embeddings would risk placing some topics of a document in the training set and other topics of the same document in the test set. Because topics from the same document are semantically related and often highly correlated, this would allow the classifier to exploit document-specific patterns it has already "seen" in training, artificially inflating performance. To prevent this, we assign each document a unique group identifier g_j. Using Group K-Fold, all embeddings belonging to the same document are kept together, ensuring that the model is always evaluated on entirely unseen documents. This setup mirrors the intended deployment scenario of retrieval-augmented generation, where the system must generalize to new documents, not merely to new topics of documents it has partially observed during training.

All preprocessing steps (L2 normalization, SafePCA) are fit only on the training portion of each fold through the pipeline, ensuring that no statistics from the test portion influence training or tuning. We also report standard classification metrics on each outer test split (macro-F1, accuracy, macro-precision, macro-recall) for completeness.

RETRIEVAL SCENARIOS (EVALUATED ON OUTER TEST SPLITS)

For each outer test fold and each class $c \in \mathcal{Y}$, we evaluate two ranking strategies over test chunks:

Cosine Baseline (class-agnostic) as used in RAG

- Score every test item $\tilde{x}$ by cosine similarity to the fixed class query $\widetilde{q_c}$: $s_{\cos}(\tilde{x}; c) = \widetilde{q_c}^{\top}\tilde{x}$.
- Rank all test items by $s_{\cos}$ in descending order.

Known-Label, Classifier-Probability Ranking.

- Assuming the user's information need specifies class c, we rank test items by the learned posterior for c: $s_{\text{clf}}(x; c) = p(y = c \mid x;\ \theta^{\star})$, where $\theta^{\star}$ are the inner-Cross Validation–selected parameters trained on the outer-train split. We compute $p(y \mid x)$ via Linear Regression's softmax or the calibrated SVM probabilities, after the pipeline's transform (L2 ± SafePCA). Items are sorted by $s_{\text{clf}}(\cdot; c)$ descending.

EVALUATION METRICS

Let π_c denote the permutation (ranking) of test indices produced by a method for class c. Let $\mathcal{R}_c$ be the set of relevant indices (test items with true label c).

$$Hits@k = \sum_{t=1}^{k} 1\{\pi_c(t) \in \mathcal{R}_c\}\ for\ k \in \{1,3,5\}$$

Here, $t$represents the rank position within the retrieval list (with $t = 1$ being the highest-ranked item) $\pi_c(t)$ is the t-th retrieved item for class c, and $1\{\pi_c(t) \in \mathcal{R}_c\}$ is an indicator function that equals 1 if the retrieved item is relevant and 0 otherwise. Thus, Hits@k quantifies the number of relevant items retrieved within the top-k positions for a given class.

$Mean\ Reciprocal\ Rank(MRR)@k$: Let r_c be the (1-based) rank position of the first relevant item, truncated to k (if none within top-k, set reciprocal to 0). Formally,

$$MRR@k\ = \begin{cases} \frac{1}{r_c}, if\ \exists\ t \le k\ s.t. \pi_c(t)\ \in\ \mathcal{R}_c\ and\ r_c = min\{t \le k : \pi_c(t)\ \in\ \mathcal{R}_c\}, \\ 0, otherwise. \end{cases}$$

Effort to Full Recall $k_{\text{recall}=1}$ which captures the average number of items a user must inspect to achieve full recall of relevant results:

$$k_{\text{recall}=1} = \min\{k: \{\pi_c(1), \ldots, \pi_c(k)\}\ \supseteq\ \mathcal{R}_c\}$$

If $|\mathcal{R}_c| = 0$ in a fold, that (fold, class) is skipped for retrieval metrics.

LIFT COMPUTATION AND AGGREGATION

For each (fold, class) we compute lifts comparing classifier-probability ranking against the cosine baseline:

- $Lift_{hits@k} = \begin{cases} \frac{hits@k_{clf}}{hits@k_{cos}}\ if\ hits@k_{cos}\ >\ 0, \\ NaN\ otherwise. \end{cases}$

- $Lift_{MRR@k} = \begin{cases} \frac{MRR@k_{clf}}{MRR@k_{cos}}\ if\ MRR@k_{cos}\ >\ 0, \\ NaN\ otherwise. \end{cases}$

- $Lift_{k_{recall=1}} = \begin{cases} \frac{k_{cos}}{k_{clf}}\ if k_{cos}\ >\ 0, k_{clf}\ >\ 0, \\ NaN\ otherwise. \end{cases}$

Values > 1 indicate improvement (Hits@k: more relevant within fixed k, MRR@k: earlier first relevant, Effort to Full Recall: fewer items to achieve full recall).

Unless otherwise stated, we report macro summaries as the mean of per-(fold,class) ratios (mean-of-ratios), which treats each class and fold equally. This differs from the ratio of means and may produce different numeric values.

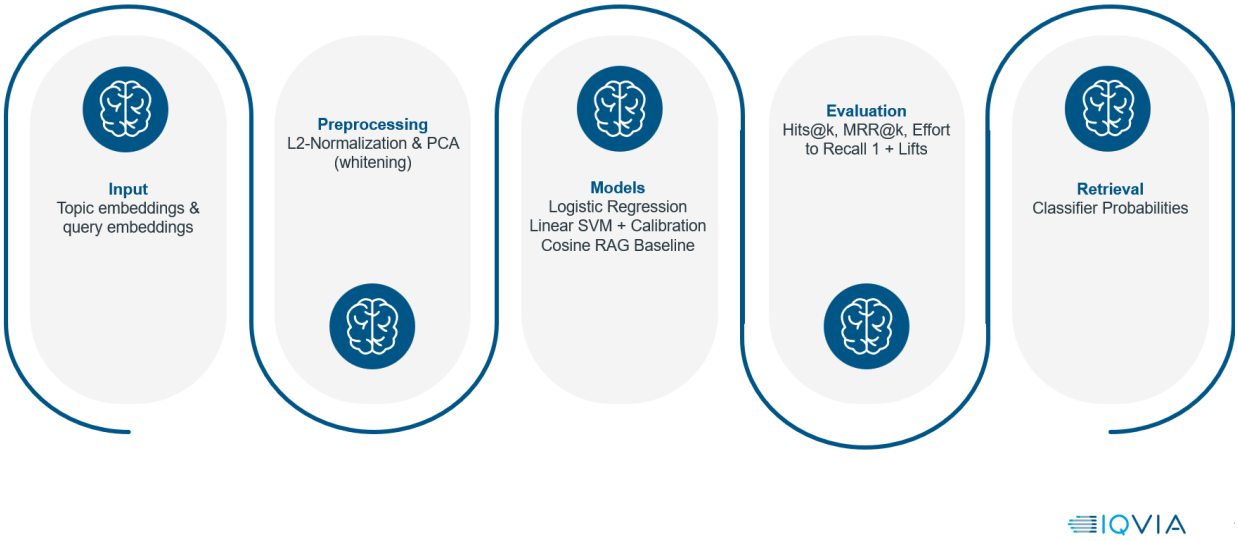

Figure 2: Overview of the Topic Classifier pipeline

TOPIC CLASSIFIER RESULTS

We compared linear classifiers (Support Vector Machine, Logistic Regression) against a cosine similarity baseline (RAG-style retrieval) for section-level classification and retrieval across the three target labels: Intervention, Objective, and Patient characteristics. As shown in Table 3, both SVM and Logistic Regression achieved consistently high accuracy across all metrics.

Model	Hits@1/Lift	Hits@3/Lift	Hits@5/Lift	MRR@1/Lift	MRR@3/Lift	MRR@5/Lift	Effort@Recall 1/Lift
SVM	**1.0 / 1.0**	**3.0 / 1.57**	**4.67 / 1.97**	**1.0 / 1.0**	**1.0 / 1.21**	**1.0 / 1.4**	**5.6 / 1.95**
LR	1.0 / 1.0	3.0 / 1.57	4.67 / 1.96	1.0 / 1.0	1.0 / 1.21	1.0 / 1.4	5.7 / 1.91
Cosine (RAG)	0.8	2.1	3.07	0.8	0.86	0.87	10.5

Table 3: Performance comparison between linear classifiers (SVM and Logistic Regression) and the Cosine Similarity baseline. Best performing model is shown in **bold**.

As shown in Table 4, for Hits@k and MRR, the linear models retrieved relevant topics reliably, with near-perfect scores at k=1, 3, and 5. For example, SVM achieved Hits@1 = 1.0 across all labels, with corresponding MRR@1 = 1.0, indicating that the correct section was ranked first in every test fold. Logistic Regression exhibited similarly strong results, with slight variation only in Effort@Recall1. In contrast, the cosine baseline performed notably worse. For Intervention, Hits@1 dropped to 0.4, and Effort@Recall1 more than doubled relative to SVM and LR (13.0 vs ~5.5–6.0). Even for Objective and Patient characteristics, where cosine similarity fared better, performance lagged behind linear models in both precision of early retrieval (Hits@1, Hits@3) and efficiency of full recall.

To quantify improvement over cosine similarity, we computed lift values for each metric (Table 3). Both SVM and Logistic Regression demonstrated substantial gains. For Hits@3, lifts of 1.57 were observed, while for Hits@5 lifts approached 2.0, indicating that the linear models retrieved nearly twice as many relevant items within the top 5 ranks compared to the cosine baseline. Improvements were also evident in ranking quality: MRR@3 lifts of ~1.2 and MRR@5 lifts of ~1.4 were achieved. Effort@Recall1 was nearly halved, with lift values around 1.9–1.95 for SVM and LR, reflecting reduced burden in retrieving all relevant items.

Model	Label	Hits@1	Hits@3	Hits@5	MRR@1	MRR@3	MRR@5	Effort@Recall1
SVM	Intervention	1.0	3.0	4.6	1.0	1.0	1.0	**5.6**
SVM	Objective	1.0	3.0	4.8	1.0	1.0	1.0	**5.2**
SVM	Patient characteristics	1.0	3.0	4.6	1.0	1.0	1.0	6.0
LR	Intervention	1.0	3.0	4.4	1.0	1.0	1.0	**6.0**
LR	Objective	1.0	3.0	4.8	1.0	1.0	1.0	**5.2**
LR	Patient characteristics	1.0	3.0	4.8	1.0	1.0	1.0	5.8
Cosine (RAG)	Intervention	0.4	1.2	2.2	0.4	0.57	0.62	**13.0**
Cosine (RAG)	Objective	1.0	2.2	2.6	1.0	1.0	1.0	**12.8**
Cosine (RAG)	Patient characteristics	1.0	3.0	4.4	1.0	1.0	1.0	5.8

Table 4: Performance comparison between linear classifiers (SVM and Logistic Regression) and the Cosine Similarity baseline across topics.

Overall, the results demonstrate that linear classifiers substantially outperform cosine similarity for topic classification and retrieval in scarce-data settings. Both SVM and Logistic Regression offer strong, stable performance across all labels, while cosine similarity suffers particularly on Intervention queries. The use of class-specific probabilistic models not only improves early precision (Hits@1, MRR@1) but also reduces retrieval effort, validating the design choice of integrating supervised classifiers into the RAG pipeline.

OUTLINE GUIDED GENERATION

MOTIVATION

Automated generation of regulatory text faces two recurring problems. The first is uncontrolled content expansion, where large language models (LLMs) produce verbose or tangential passages that exceed the intended scope of a section. The second is prompt brittleness in information extraction from PDFs: prompts that SMEs design to work on one document rarely generalize to others, forcing continual re-engineering of instructions and limiting scalability. These challenges increase the cost of automation and undermine reproducibility.

PIPELINE

To address these issues, we designed an example-based text generation pipeline (see Figure 3) that anchors both extraction and drafting to editable outlines. The process begins with outline generation, where the LLM analyses exemplar documents and proposes a draft section outline. This outline encodes the expected structure, including headings, subpoints, and evidence types, as well as stylistic instructions such as tone, language, and target audience.

The draft outline is then refined by SMEs in an editing stage, ensuring that the section plan is scientifically and regulatorily appropriate. Once finalized, the outline is transformed into a structured dictionary of variable–description pairs that need to be extracted from the source documents to generate the text as described in the outline. This schema replaces manual prompt engineering by automatically generating extraction instructions that can be applied across PDFs.

Using this schema, the LLM performs value extraction from the relevant source documents. Each extracted item is anchored to its provenance at the level of the source file from which it originates, enabling traceability, referencing and human verification. In the final stage, the LLM combines the SME-edited outline with the extracted values to produce a draft section. Because generation is guided by outline constraints and grounded in provenance-linked evidence, the resulting text is controlled in scope, consistent in style, and regulator-ready.

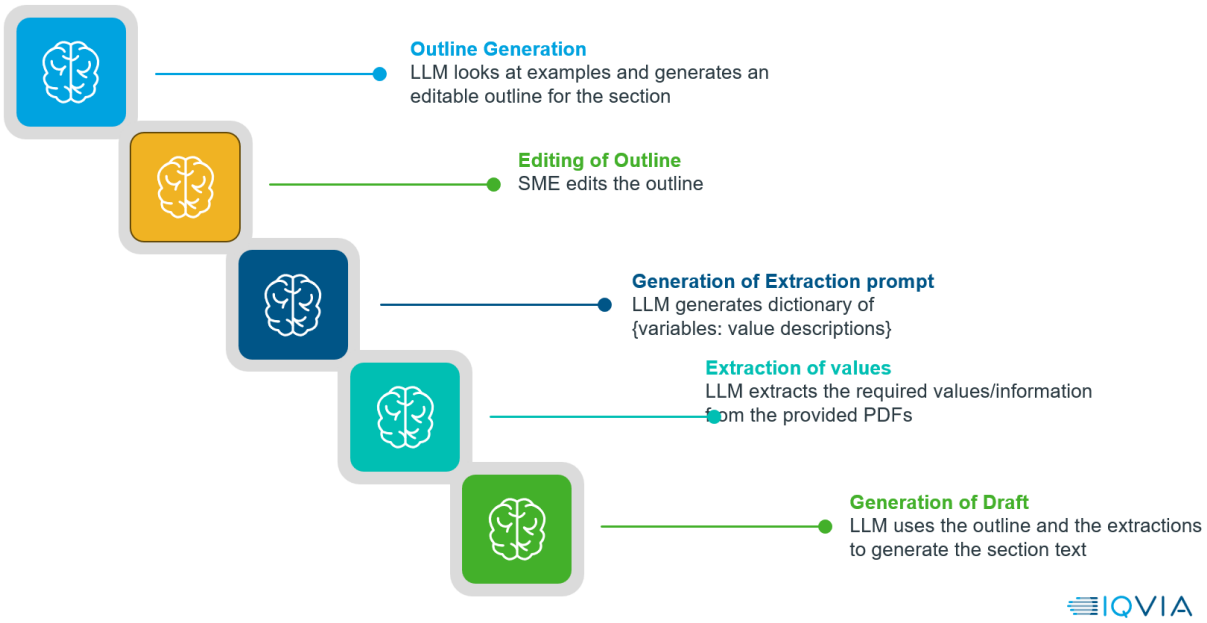

Figure 3: The overview of the Outline Guided Generation Pipeline

ADVANTAGES

This example-based approach offers several benefits. By constraining generation through outlines, it prevents uncontrolled expansion and enforces alignment with dossier templates. By automating the generation of extraction schemas, it reduces prompt brittleness and removes the need for SMEs to tailor prompts to individual PDFs. Finally, by linking every extracted item to its provenance, the pipeline ensures auditability and supports human-in-the-loop verification. Overall, the approach shifts expert effort from repetitive prompt engineering to high-level outline editing, yielding a more scalable, reproducible, and transparent solution for LLM-assisted medical writing.

CONCLUSION

This study demonstrates that the automation of SLR and HTA authoring can be made both reliable and reproducible when large language models are integrated into domain-specific, error-aware pipelines. By systematically addressing key challenges - such as fragmented table reconstruction, retrieval imprecision, extraction prompt brittleness and uncontrolled text generation - the proposed framework achieves substantial improvements in extraction fidelity, retrieval performance, and stylistic consistency. The combination of structure-aware parsing, topic-guided retrieval, and outline-constrained extraction and drafting provides a scalable blueprint for deploying LLMs in evidence synthesis workflows while maintaining traceability, auditability, and regulatory compliance. While expert oversight remains indispensable, these advances significantly reduce manual burden and establish a foundation for future work focused on multimodal reasoning, adaptive retrieval optimization, and automated extraction and drafting to further align machine-generated outputs with the standards of expert medical writing.

REFERENCES

[1] OpenAI, “GPT-4o System Card,” May 2024. [Online]. Available: https://openai.com/index/gpt-4o-system-card/ . [Accessed: 29-Oct-2025].

[2] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[3] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[4] A. Ng and M. Jordan, “Feature selection, L1 vs. L2 regularization, and rotational invariance,” in *Proc. 21st International Conference on Machine Learning (ICML)*, Banff, Canada, 2004, pp. 78-85.

[5] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, Sept. 1995.

[6] J. C. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods,” in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf & D. Schuurmans, Eds., MIT Press, 2000.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Ioannis Reklos
Company: IQVIA Technologies
Address: 284 Kifisias Avenue, 152 32 Halandri, Greece
Work Phone: +30 6945853523
Email: ioannis.reklos@iqvia.com

Brand and product names are trademarks of their respective companies.