Guidance Note

Balancing Innovation and Rigor: Guidance for the Thoughtful Integration of Artificial Intelligence for Evaluation

5/13/2025

Summary

Within the evolving landscape of artificial intelligence, large language models (LLMs), a type of generative artificial intelligence, offer significant potential for improving the collection, processing, and analysis of large volumes of text data in evaluation. In this note, we present key lessons and good practices for leveraging LLMs based on our recent experiments. The experiments' results reveal that the LLMs tested could perform text classification quite well, achieving satisfactory recall, precision, and F1 scores. The models also performed well on tasks such as text summarization and synthesis, achieving high scores on metrics related to relevance, coherence, and faithfulness of the generated text. However, challenges remain in ensuring completeness and relevance in information extraction and text synthesis tasks. We found iterative prompt validation and refinement, measurement of model performance with relevant metrics, and representative sampling to be important considerations to ensure the success of these applications. We hope this document will serve as a practical resource for multidisciplinary teams across evaluation departments seeking to responsibly integrate LLMs into their workflows by maintaining analytical rigor.

Keywords

Artificial intelligence; data science; evaluation; generative artificial intelligence; large language model; natural language processing.

This publication was jointly produced by the Independent Evaluation Group (IEG) of the World Bank (WB) and the Independent Office of Evaluation (IOE) of the International Fund for Agricultural Development (IFAD).

Contents

Figures

Tables

Key Takeaways

Identify relevant use cases. Thoughtful experimentation begins with identifying evaluation methods in which LLMs can be integrated to add significant value compared with traditional approaches within the same resource constraints. Leveraging LLMs will not be suitable for every use case; therefore, it is essential to align experiments with those use cases where LLMs' capabilities can be leveraged effectively.

Plan workflows within use cases. Breaking down use cases into detailed steps and tasks helps teams understand where and how to apply LLMs effectively. This modular design also allows for the reuse of successful components within and across use cases.

Understand and agree on resource allocation and outcomes. Teams must clearly understand and agree on the necessary resources and expected outcomes for an experiment. This includes human resources (evaluator, data scientist, research design and domain experts), technology, timeline, and a definition of success for each experiment.

Form an appropriate sampling strategy. A robust sampling strategy is essential, such as dividing a data set into training, validation, testing, and prediction sets to facilitate effective prompt development and model evaluation. Such division can help a team refine prompts iteratively and assess their generalizability, ultimately leading to more aligned responses from LLMs.

Select appropriate model evaluation metrics. Selecting and calculating metrics to measure LLM performance, along with appropriate intercoder reliability assessments for human-annotated data, is crucial to determine the success of an experiment. For discriminative tasks such as text classification, standard machine learning metrics such as recall, precision, and F1 scores can be useful. For generative tasks such as text summarization and synthesis, human assessment criteria such as faithfulness, relevance, and coherence can be meaningful.

Iteratively develop and validate prompts. Developing effective prompts involves iteratively testing and refining. For example, a team could start with a basic prompt and gradually add more specific instructions based on LLMs' responses. Including requests for justification in prompts can provide insights into a model's reasoning and help with prompt refinement.

Abbreviations

AI	artificial intelligence
GenAI	generative artificial intelligence
IEG	Independent Evaluation Group
LLM	large language model
SLR	structured literature review

All dollar amounts are US dollars unless otherwise indicated.

Acknowledgments

This guidance note was authored by Harsh Anuj, Hannah Den Boer, and Estelle Raimondo. Dawn Roberts, Jenny Gold, Mercedes Vellez, and Joy Butscher collaborated with the authors on the experiments. Jenny Gold and Ridwan Bello provided helpful comments on an earlier draft. Arunjana Das, Amanda O'Brien, Wendy Rubin, and William Stebbins, assisted with the editing, production, and dissemination of the guidance note. The authors are grateful to Sabine Bernabè and Dr. Indran A. Naidoo for their support. Microsoft Copilot was leveraged during the production of this document.

Introduction

Within the evolving landscape of artificial intelligence (AI), large language models (LLMs)—a type of generative artificial intelligence (GenAI) for text (see Brown et al. 2020; Google 2025)—have the potential to enhance the efficiency, breadth, and validity of the collection, processing, and analysis of text as data in evaluation practice (see Raimondo et al. 2023a, 2023b, 2023c; Ziulu et al. 2024; Anuj et al. 2025).[1] However, LLMs do not always generate aligned, authoritative, or accurate responses (see Ouyang et al. 2022; Martineau 2023; OpenAI 2024), indicating that their responses must be validated before use in our work. Furthermore, the importance of analytical rigor in our practice, combined with our institutions' ability to affect the lives of people around the world, makes it clear that we must take a thoughtful approach to integrating such tools.

How can we realize the potential of LLMs while maintaining rigor? This guidance note aims to answer that question by demonstrating good practices for experimenting with LLMs based on a frequently occurring use case in our evaluations: structured literature review (SLR). This use case serves as a concrete example of how LLMs can be thoughtfully integrated into evaluation workflows.

Our findings are based on a series of on-the-job experiments conducted by the Independent Evaluation Group (IEG) over a two-month period in late 2024. These experiments were carried out within a multidisciplinary team comprising IEG and International Fund for Agricultural Development staff with expertise in evaluation, data science, and research design.

In the next section, Key Considerations for Experimentation, we describe how to identify relevant use cases and opportunities within use cases for the application of LLMs, the importance of finding agreement on resources and outcomes, and the selection of appropriate metrics to measure LLM performance. The section includes a detailed workflow for an SLR, while the workflow for an evaluation synthesis is presented in the appendix, along with a more "traditional" SLR workflow. The section Our Experiments and Results presents the design and results of our experiments and includes tables summarizing the performance of LLMs on text classification, summarization, synthesis, and information extraction, as measured by selected metrics. The next section, Emerging Good Practices, offers guidance for developing effective prompts, creating subsets of data to compute model evaluation metrics, and refining prompts based on validation findings. Finally, in the last section, Going Forward, we discuss the ongoing journey of

[1] Some LLMs such as OpenAI's GPT-4o are inherently multimodal—that is, they can accept and or generate images, speech, or other types of data along with text. See for example Huyen 2023 for a helpful description of multimodality.

experimentation with AI in evaluation offices, emphasizing continuous learning, adaptation, and collaboration.

Key Considerations for Experimentation

Based on our experience, we identified the following key considerations to assess the potential for thoughtful integration of LLMs in use cases related to evaluative analyses and syntheses.

Identifying Use Cases

Thoughtful experimentation begins with careful planning and the identification of areas in which LLMs could add *sufficient incremental value* for a given set of resources and constraints (for example, staff, budget, time) compared with more traditional approaches to the analysis of text data. This foundational step ensures that experiments are purposeful and relevant. Although LLMs are quite versatile and seemingly all-knowing, their usefulness depends on the way they are applied for particular use cases. Misaligned experimentation risks wasting resources and compromising quality.

Such use cases typically meet the following conditions: (i) The literature on LLMs (and or previous work) identifies the case as having high-value applications, such as text classification, text summarization, sentiment analysis, and information retrieval (see Puri et al. 2019; Lewis et al. 2020; Gera et al. 2022; Alaofi et al. 2024; Glickman et al. 2024); and (ii) the current evaluation practice is either inefficient, 'shallow', or impossible due to the sheer volume of text.

For this guidance note, we built on the eight limited experiments on applications of LLMs for evaluation practice that we had carried out and published as a series of blogs (Raimondo et al. 2023a, 2023b, 2023c). We chose to focus on one of the two use cases that had yielded unimpressive results: SLRs. We also examined the other use case that had not worked well: evaluation synthesis. We expect LLMs to enhance the way in which these two important methods for our major evaluations are implemented.

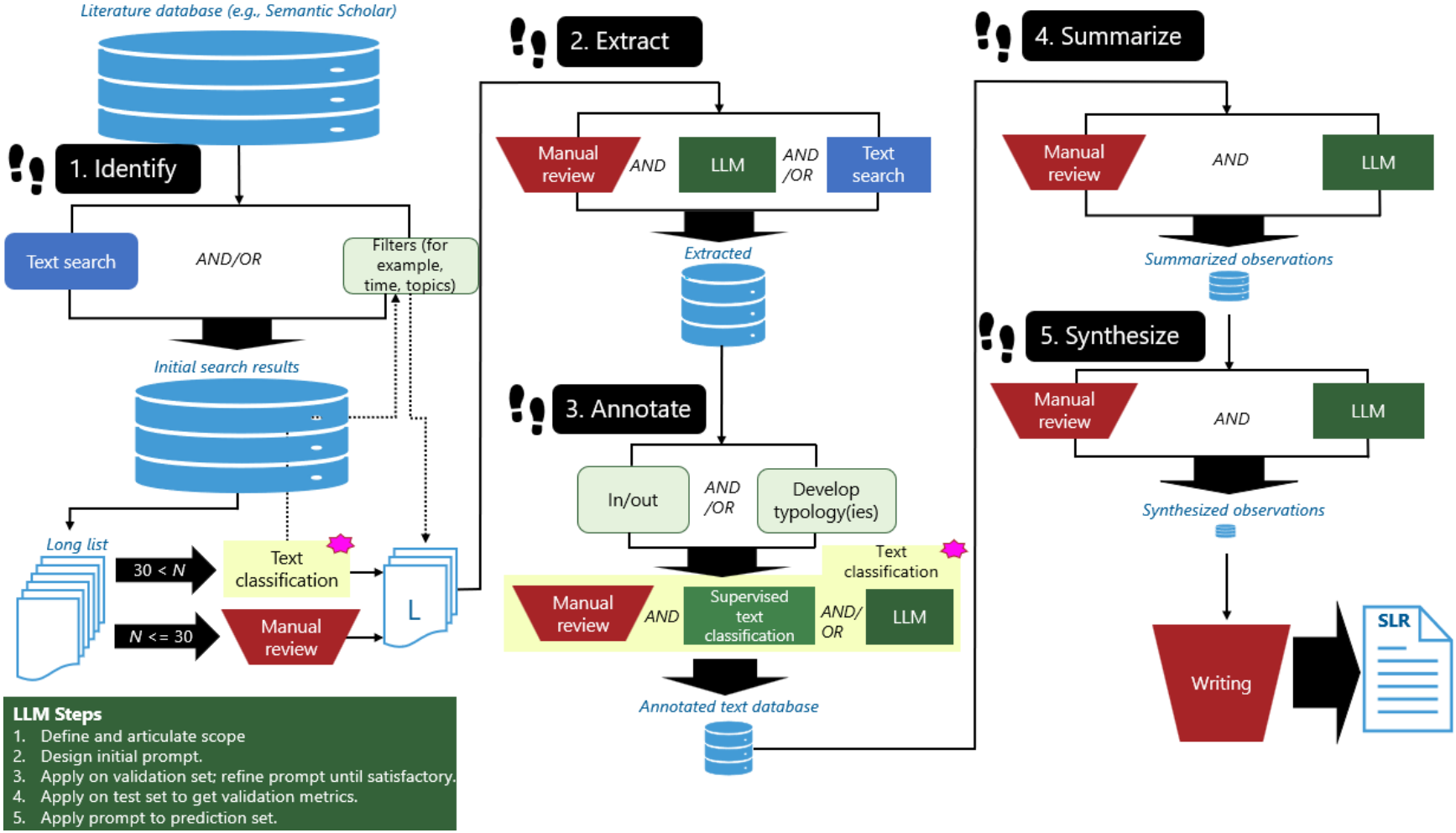
Identifying Opportunities Within Use Cases

We learned from previous experiments that for complex use cases such as SLRs, it is important to unpack the various analytical steps and to carefully examine for what and how LLMs can be leveraged. This step requires the development of a granular understanding of the analytical steps involved, as well as the capabilities of LLMs. Although it is tempting, for example, to try to produce an SLR or evaluation synthesis with a few documents and simple prompts, we had found this approach to be unsuccessful.

Therefore, we started by creating a relatively detailed workflow—through a data science lens—for the various steps in the selected use cases. (For reference, we have provided a visualization of the workflow for a standard SLR as per IEG 2017 in the appendix.) In doing so, we found that the workflows for the two use cases (as well as other ones that interest us, such as portfolio review and analysis and interview transcript analysis) are broadly very similar. We also noticed that within the steps in the workflow, specific components can be repeated and provide opportunities to use the capabilities of LLMs that we know can work well (based on the literature and our previous applied work and experiments). These capabilities include (i) text classification, (ii) text summarization, (iii) text synthesis, and (iv) summative information extraction.

The workflow for SLR is provided in figure 1, and a workflow for the evaluation synthesis is provided in figure A.2. Both figures show that components such as text search, manual review, text classification, and LLM appear multiple times within and across workflows. This modularity is by design and assists with the identification of task-specific opportunities for successfully applying LLMs. (In practice, the steps can overlap because the process is iterative, with multiple feedback loops.) The modularity can also be helpful when developing similar workflows for other use cases, such as portfolio review and analysis and analysis of interview transcripts, which we are currently implementing at IEG. From a developer's perspective, this modularity is helpful when developing Python code to semi-automate various steps with humans in the loop. It is important to note that the manual review component is mandatory in our workflows when LLM or machine learning are used.

Figure 1. Structured Literature Review Workflow



Source: Independent Evaluation Group.
Note: LLM = large language model; SLR = structured literature review.

Figure 1 also shows that there are five moments that present opportunities to leverage LLMs: (i) When screening documents for inclusion in the review or synthesis based on their relevance to the topic; (ii) when extracting relevant information from documents; (iii) when annotating extracted text to various typologies; (iv) when summarizing annotated text within types; and (v) when synthesizing annotated text across types.

Finding Agreement on Resources and Outcomes

After completing the task of developing a clear road map for the application of LLMs in the use cases, team members need to harmonize expectations. Our experience shows it is important for all the team members to understand the types and amounts of resources required to undertake the experiments, and to arrive at a clear collective agreement on expected outcomes or what success would look like. This agreement is especially crucial given the multidisciplinary nature of the teams carrying out such experiments. Coming to a shared agreement on resources and outcomes can also help with dispelling or at least tempering the notion that working with LLMs is straightforward and inexpensive and will produce phenomenal results each time.

In terms of types of resources, it is important to consider the availability of full-time staff, including data scientists, evaluators, subject matter experts, and research design specialists. The technology needed to carry out the work should be identified and acquired, including compute to efficiently process large volumes of data, and budget to use proprietary LLMs via their respective application programming interfaces (APIs).[2]

Finally, it is important to define the expected outcomes from the use of LLMs, including what would be considered a successful or helpful application. Expected outcomes should be commensurate with the resources allocated. For example, in our application of an LLM in the identification step of an SLR, we agreed to consider it a success because the process allowed us to identify (via a semantic search), bulk download, and screen the full text of over 10,000 research papers for relevance in a short duration and with an acceptable level of accuracy (see Selecting Appropriate Metrics to Measure LLMs'

[2] To learn more about compute, see Amazon Web Services (n.d.-b). To learn about APIs, see Goodwin 2024. In our experiments, we used OpenAI's proprietary GPT-4o model via their API as well and playground. Access to compute, especially sophisticated NVIDIA graphics processing units (GPUs), is necessary for using open-source models directly. We conducted some tests with open small models from Mistral AI, Microsoft, Google, and Meta, but due to our limited access to graphics processing units at the time, we could not test the larger models that might be able to compete with GPT-4o. However, the cost for GPT-4o was not insignificant, and free, open models with similar performance would certainly be a strong choice going forward, for a variety of reasons, given that they can be securely integrated into an institution's information technology systems.

Performance). This application made the process significantly more efficient and comprehensive than a purely manual one would have been, while *reducing* the overall effort required.

Selecting Appropriate Metrics to Measure LLMs' Performance

While the criteria for assessing whether an experimental application of LLMs for an evaluation use case is successful or not are subjective, it is important to think about clear dimensions to measure LLMs' performance on more narrowly defined tasks, such as text classification, summarization, synthesis, and information extraction.

Continuing with the SLR example, use of an LLM for literature identification (classification) with a recall score of 0.75 with precision score of 0.6 could be considered a success in one evaluation, whereas in another evaluation recall and precision scores of 0.9 and 0.5 respectively might be considered successful. However, to establish whether the applications were successful, the recall and precision scores need to be selected and computed first.

For the text classification task in our experiment, we leveraged standard machine learning model evaluation metrics such as binary classification accuracy, recall, precision, balanced accuracy, and F1 scores respectively. These metrics measure the degree of overlap between machine-annotated "predicted" labels and human-annotated "ground-truth" labels.[3] Furthermore, we split the underlying samples of papers into distinct training, validation, testing, and prediction sets respectively. As is standard practice in machine learning, the testing set was not used to develop or refine the prompts or other inputs to the process, which enabled us to compute unbiased estimates for our selected performance metrics with it (see Emerging Good Practices below).

However, for the text summarization, synthesis, and information extraction tasks, we did not develop a human benchmark to use for assessing responses. This was because we did not apply these tasks to a real evaluation use case, and therefore did not have the resources to produce human-annotated data.[4] In the absence of a directly comparable "ground truth," how can we assess the quality of model responses? We used the following criteria—faithfulness, relevance, and coherence—which can provide comprehensive and accurate feedback as they allow for a subjective assessment of the

[3] When developing the "ground-truth" labels, it is important to take intercoder reliability into account.

[4] A human-generated reference text also offers the option to leverage relevant model evaluation metrics for natural language generation such as BLEU, METEOR, and ROUGE.

generated texts' alignment with an evaluation task's objective and an evaluator's expectations.

i. Faithfulness measures whether the information generated is factually consistent with the information in the source or not (see Durmus et al. 2020; Zhang et al. 2024).

- Relevance measures whether the selected content from the source is the most important content following the prompt (see Fabbri et al. 2021; Zhang et al. 2024).
- Coherence measures the overall collective quality of the sentences: The response text should be built from sentence to sentence to a coherent body of information about a topic (see Fabbri et al. 2021; Zhang et al. 2024).

Table 1 provides details on the above criteria. To determine what minimum values for each metric would be acceptable for the application to be considered a success, we took a context-specific approach. For literature identification (a classification task), recall and precision scores higher than 0.6 and 0.7, respectively, were deemed necessary. This was due to two factors: (i) the conceptual complexity of classification task due to the complexity of the SLR topics, and (ii) the class imbalance in the underlying search results from the Semantic Scholar open data platform (Kinney et al. 2023).[5] Similarly, users can determine what values of the metrics measuring faithfulness, relevance, and coherence would be satisfactory for their tasks. For use cases with higher stakes (where, for example, a real-world decision must be made using LLMs' responses, even in part), higher values would be required.[6] Finally, it is important to note that human judgments on Likert scales can vary; therefore, it is recommended that evaluators measure and report interrater agreement through a metric such as Cohen's kappa (see McHugh 2012).

Table 1. Assessment Criteria

Criterion	Definition	Assessment Scale	Source(s)	Task[a]
Faithfulness	Being factually consistent with information in the source document	0 (unfaithful) or 1 (faithful) If 0, then H[b] or IC[c]	Durmus et al. 2020; Zhang et al. 2024	Summarization, synthesis, extraction,
Relevance	Selection of important content from the source document	Likert scale of 1–5	Fabbri et al. 2021; Zhang et al. 2024	Summarization, synthesis,

[5] That is, the results from Semantic Scholar bulk search API contained a high proportion of false positives. This was an intended outcome of our strategy for the initial search. We kept our search terms relatively broad to maximize recall (see IEG, Forthcoming).

[6] Only the text classification task was used for an evaluation, so no practical thresholds were set in advance for the text summarization, text synthesis, and information extraction use cases, as these were applied to purely experimental tasks.

Criterion	Definition	Assessment Scale	Source(s)	Task[a]
		(1 = highly irrelevant, 5 = highly relevant)		extraction
Coherence	Collective quality of all sentences	Likert scale of 1–5 (1 = highly incoherent, 5 = highly coherent)	Fabbri et al. 2021; Zhang et al. 2024	Summarization, synthesis
Binary classification accuracy score	Fraction of correct classifications	0 (completely inaccurate) to 1 (completely accurate)	Pedregosa et al. 2011	Classification
Precision score	TP / (TP + FP)[d]	0 (completely inaccurate) to 1 (completely accurate)	Pedregosa et al. 2011	Classification
Recall score	TP / (TP + FN)[d]	0 (completely inaccurate) to 1 (completely accurate)	Pedregosa et al. 2011	Classification
Balanced accuracy score	Arithmetic mean of sensitivity (true positive rate) and specificity (true negative rate)	0 (completely inaccurate) to 1 (completely accurate)	Pedregosa et al. 2011	Classification
F1 score	Weighted harmonic mean of precision and recall scores	0 (completely inaccurate) to 1 (completely accurate)	Pedregosa et al. 2011	Classification

Source: Independent Evaluation Group; Pedregosa et al. 2011; Durmus et al. 2020; Fabbri et al. 2021; Zhang et al. 2024.
Notes:
a. Multiple metrics can, and should, be combined to assess the results of a particular task, as discussed earlier in this section.
b. H = hallucination (that is, information expressed is not contained in the source).
c. IC = incorrect concatenation (that is, information expressed conflicts with the source).
d. TP = true positive; FP = false positive; TN = true negative, FN = false negative.

Our Experiments and Results

Given the reuse of components within and across the SLR and evaluation synthesis workflows—as well as the resources and time required to undertake such experiments in practice for a major evaluation at IEG—we did not conduct experiments for the full SLR workflow or the evaluation synthesis workflow. Instead, we focused on robustly testing the components of the literature identification step, including LLM-based text classification, for an SLR in an ongoing IEG thematic evaluation of the World Bank Group's support for epidemic preparedness (World Bank, forthcoming). We then used random samples from identified literature to conduct experiments with text summarization, text synthesis, and information extraction. Table 2 provides details on the design of our experiments.

Table 2. Our Four Experiments

Item	Task	Sample	Model response	Unit of Scoring	Model and Parameters
Text classification	Binary classification to identify literature on private sector engagement in epidemic preparedness	30 papers in test set. Selected via text clustering	Categorization and justification for each paper	Each categorization response	OpenAI GPT-4o model via API Temperature = 0.0
Text summarization	Generation of abstracts from full papers	30 papers. Selected randomly from search results	Abstract for each paper	Each generated abstract	OpenAI GPT-4o-mini model via API Temperature = 0.0
Text synthesis	Generation of a synthesis from six summaries on private sector engagement in epidemic preparedness	Six summaries of 200 words each. Selected randomly from text summarization results	One 500-word synthesis	Each of the five paragraphs. Each paragraph included the pattern, the examples, and a conclusive overarching sentence.	OpenAI GPT-4o mini via playground Temperature = 0.0
Information extraction	Extraction of information on public–private sector engagement in epidemic preparedness contained in papers. Three types of information were to be extracted: actors, mechanism, and goals	12 papers. Selected from validation set used in one of the text classification tasks. Selected randomly	57 responses returned (three categories for 19 examples, as one paper could contain multiple examples).	Each response per paper (that is, three responses per paper)	OpenAI GPT-4o-mini model API Temperature = 0.0

Source: Independent Evaluation Group.
Note: API = application programming interface.

Tables 3 and 4 summarize the results for each experiment.

Table 3. Experiment Results for Discriminative Task

Task\Score	Accuracy	Recall	Precision	F1	Balanced Accuracy
Text classification (testing set)	0.90	0.75	0.60	0.67	0.67

Source: Independent Evaluation Group.
Notes: We assume here that a 'discriminative task' is one for which the required response is in the form of a decision regarding the appropriate category for an observation. See also entry for *discriminative models* in Google [2025] for a definition.

Table 4. Experiment Results for Generative Tasks

Task\Score	Faithfulness (IC)	Faithfulness (H)	Relevance	Coherence
Text summarization	0.90	1.00	4.87	4.97
Text synthesis	1.00	1.00	4.20	5.00
Information extraction	1.00	1.00	3.25	n.a.

Source: Independent Evaluation Group.

Notes: We assume here that 'generative tasks' are those for which the required response from a model is in the form of a narrative. See also entry for *generative model* in Google [2025] for a definition (or lack thereof). n.a. = not applicable because the responses only included one sentence. IC = incorrect concatenation (that is, information expressed conflicts with the source); *H* = hallucination (that is, information expressed by the model is not contained in the reference text).

As can be seen in tables 3 and 4, the LLMs we tested generally performed quite well in each of the generative tasks based on the metrics used. The model responses for the text summarization task were remarkably relevant, coherent, and faithful. The high relevance score (4.87) shows that the abstracts generated contained the most important information, often outperforming original abstracts where those were present. A coherence score of 4.97 highlights the ability to produce unified, logically connected responses, whereas a faithfulness score of 0.90 reflects strong factual alignment, with only some isolated issues with incorrect aggregation of information. Importantly, no hallucinations were observed. For the information extraction task, faithfulness was excellent: Information was accurately retrieved (faithfulness incorrect concatenation [IC] = 1.00), and no hallucinations took place (faithfulness hallucination [H] = 1.00). However, the relevance score (3.25) shows that the model had difficulty extracting the most relevant information from the papers, particularly in identifying specific requested details, and omissions of relevant information were noted. In the text synthesis task, which was a summary of summaries, information was accurately retrieved (faithfulness IC = 1.00), and no hallucinations took place (faithfulness H = 1.00).[7] Additionally, the LLMs correctly referenced over 10 times the number of respective summaries that it had used to produce the 500-word synthesis, as we had stipulated in the prompt. However, some relevant information was omitted, hence the lower relevance score of 4.20.

The text classification task yielded strong results after multiple iterations to refine the prompt using the validation set. Given the complexity of the task owing to the topics of the literature review, the need to keep overfitting in check,[8] and the efficiency introduced by the overall workflow, the recall and precision scores of 0.75 and 0.60 respectively were deemed satisfactory in this particular use case (see Liu et al 2018). Indeed, the use of the same workflow and prompt format for different SLR subtopics

[7] Because the synthesis was conducted with summaries of the source documents, the results were likely better compared with what we might have achieved by synthesizing the source texts directly.

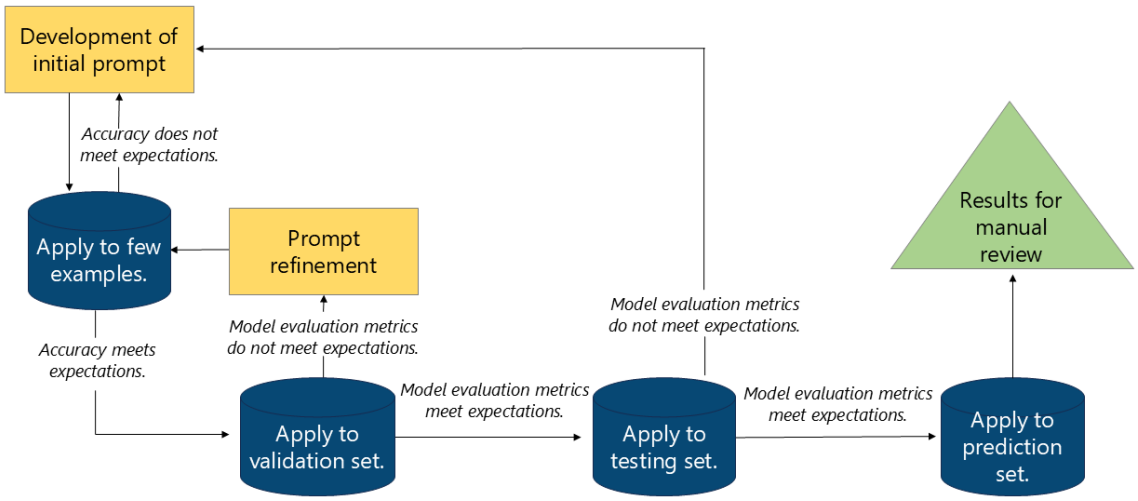
[8] For more information about overfitting, see Google (2024).

yielded helpful results, likely due to the use of representative sampling based on a semi-supervised learning strategy (see Géron 2019; Liu et al. 2018) that supported generalizability.

Emerging Good Practices

Given that our experiments yielded satisfactory results, sometimes after a few or many iterations, we identified some good practices that helped us achieve useful results. Most of our guidance in this section focuses on the prompting and validation loop because this is an important factor for achieving satisfactory results on our selected LLM evaluation metrics (see Shin et al. 2020 for a discussion on the importance of prompting).[9] Figure 2 describes this iterative process. This guidance is based on our work on the various experiments described earlier. These practices emerged as ones that contributed to satisfactory results in this set of experiments and were identified by us either during this work or during our past work with LLMs.

Figure 2. Prompting and Validation Loop



Source: Independent Evaluation Group.

As is standard practice in machine learning, the data set on which a prompt is applied to get the desired response should first be divided into training, validation, testing, and

[9] Various steps before the application of LLMs are important and were applied in our experiments. For example, an efficient and accurate retrieval system before LLM application (see Lewis et al. 2020) and minimizing context length (see Liu et al. 2024), among others. See IEG, Forthcoming for more details on the methodology.

prediction sets, respectively.[10] The training set consists of a few human-annotated examples that are included in the prompt for the model to learn from when analyzing each unlabeled observation. The validation set consists of several human-annotated examples on which the prompt is applied, and model evaluation metrics are established. If these metrics are found to be unsatisfactory for the context of the task, then the prompt is refined until the results for the validation set are deemed satisfactory.

Then, the prompt that provided the best results on the validation set is applied on a testing set and metrics are computed once more. Further prompt refinement is not done at this stage. The values of the metrics from the testing set allow us to assess the prompt's generalizability on observations that differ from those in the validation set and provide an unbiased picture of the accuracy we can expect on the unlabeled prediction set. If the values of the metrics from the testing set are found to be unsatisfactory, then the whole exercise should be restarted, and a different set of observations should be included in the new testing set to avoid data leakage (see Mucci 2024 for more information on data leakage).

Finally, if or when the metrics for the test set are deemed satisfactory, the prompt is applied to the unlabeled prediction set. These results then need to be manually screened for relevance. Ideally, model evaluation metrics should also be computed for this set, at least for a sample of 30 randomly selected observations. This approach will give the truest assessment of the model's performance and might provide lessons to improve accuracy in future work.

Representative Sampling

As mentioned in the previous section, it is advisable to split the data set into four distinct sets before developing even an initial prompt. Taking the following steps will ensure that the model evaluation metrics help improve generalizability of the prompts on the prediction set.

First, understand the distribution of your input data. Understanding the basic nature of your input data (for example, the text of research papers returned by an initial search) can be helpful throughout the process, including for setting and managing expectations.

---

[10] The observations in the first three sets must be annotated by humans. While the first is used to provide examples to the model, the second and third sets serve as the "ground truth" against which model evaluation metrics will be calculated. This annotation requires the judgment of at least one subject matter expert. Once again, calculating measures of intercoder reliability for the human annotated dataset is important for putting LLM performance metrics into context. For example, if two human coders only agree on 80% of the labels, and an LLM achieves 75% accuracy on the labels from one annotator, we might want to accept it as good performance.

Simple characteristics such as the extent of homogeneity or heterogeneity of the input documents can be informative. For example, if there are multiple, relatively distinct, topics in the scope of your literature review, instead of trying to identify papers for all topics at once, work on one topic at a time. Or, if the topic of your review is very broad, you can use text clustering with document embeddings to identify topic clusters and split your scope into multiple topics, and work with those respectively.[11]

Second, identify and include representative observations. For example, you can identify approximately 55 of the most representative documents in your set of unlabeled documents by using purposeful sampling or document clustering.[12] We used the latter technique in our application.[13] Ask your subject matter expert to annotate these documents and include around 5 of the most representative documents as examples in the training set (that is, in the prompt), the next 20 most representative ones in the validation set, and around 30 of the next most representative ones in the testing set. The remaining unlabeled documents should be automatically assigned to the prediction set. (See Liu et al. 2018 for why this component can be helpful.)

This sampling strategy has several advantages, as it allows us to conduct model performance assessments across meaningful categories of documents[14]. First, it ensures

---

[11] For information about clustering, see OpenAI (2022).

[12] The number can be higher if your documents tend to be longer compared to the LLM's context length, and vice-versa.

[13] We first used Semantic Scholar's bulk search API (Kinney et al. 2023) to identify a long list of potentially relevant papers for each topic of the SLR. The search was performed using a set of queries with Boolean logic, developed iteratively by the team, along with filters for document type and date range etc. The Semantic Scholar results contained hyperlinks to open access full paper PDF files where available. We then scraped these files from their respective hyperlinks. Then, we split the papers into smaller chunks under the token limit of OpenAI's text-embedding-3-large model (i.e., 8,091 tokens) and retrieved the 3072-dimensional embeddings for each chunk. Then, we took the mean of the embedding vectors across each paper's chunks to arrive at document-level aggregate embeddings. We then used the scikit-learn implementation of the k-means clustering algorithm (MacQueen 1967) to cluster the documents in the 3072-dimensional embedding space. Then, we identified the documents closest to each cluster's theoretical centroids as the cluster centroid proxies and included those in our various subsets. The work was conducted using the open-source Python programming language and various user-contributed libraries. Full details of the methodology will be shared in IEG, Forthcoming.

[14] That is, the document clusters tend to group together documents that have similar semantic properties. In other words, the meanings of the words in the documents within the same cluster are similar or related to the extent that they are close to each other in the embedding space. This happens due to the richness of semantic information captured by high quality, high-dimensional text embeddings.

semantic diversity of the samples. By sampling from multiple clusters in the high-dimensional text embedding space, we ensure our model evaluation and prompt refinement spans a range of semantic contexts rather than over-representing dominant classes as might happen with random sampling with skewed data, which can lead to biased values for model performance metrics. Second, it bolsters interpretability and supports prompt refinement. Evaluating model performance across clusters reveals strengths and weaknesses of the prompt in specific types of cases, which is especially helpful when relying on prompts for classification, as it allows us to address specific issues by adjusting the prompt format and or content. Furthermore, using prototypical examples from clusters for prompt refinement can increase its effectiveness for different types of observations. Lastly, this sampling strategy also helps to avoid sampling of near-duplicate or highly similar documents.

Developing an Initial Prompt

A good prompt for an instruction-tuned LLM (see Bergmann 2024b) typically includes the following components or sections: (i) persona to be adopted by the model (for example, evaluation analyst); (ii) detailed instructions for the task the model must undertake; (iii) the relevant text with the context in which the instructions should be carried out; and (iv) requirements such as the length and format of the response. There are various community-produced resources online for how to craft the best prompts (that is, prompt engineering; see Google 2025), and best practices change as new models emerge, so we do not include general prompting tips in this guidance note and instead focus on the following specific considerations that worked well in our experiments[15].

Check the model's prompt template. Different models (and, at times, model versions) require slightly different templates for the prompts that they can understand, so make sure to check and adapt a prompt to the specific template once you have selected a model (see Amazon Web Services n.d.-a).

Break down the task into specific steps. Be explicit about the steps the model needs to undertake to follow your instructions, a technique known as chain-of-thought prompting (see Gadesha et al. 2025). For example, if you provide the LLMs with the titles and abstracts of research papers to classify based on their relevance to your SLR topic, then it would be helpful to mention in the prompt that you will give the model the titles and abstracts of the research papers and that it should first read these text fields, then compare it with the classification criteria and instructions provided to it, then make its classification decision, and then respond in the requested format.

[15] See DAIR.AI 2025 for a helpful prompt engineering guide.

Try different prompt formats. It can be worthwhile to experiment with different prompt formats before applying a prompt to the validation set for assessment and refinement. The format of a prompt refers to the types of information it includes and the order in which information is included. Both are crucial. For example, the format of our final text classification prompt for literature identification involved starting with defining the persona for the model to adopt, followed by a high-level overview of the task, then detailed instructions, then a few labeled examples, and finally the unlabeled example for the model to classify. Our template is shown in figure 3.

Include a request for justification. Due to the opaque nature of LLMs' inner workings, it is not possible for humans to interpret their "decision-making process"[16] or to understand how exactly they arrived at their responses. This challenge can be mitigated to some extent by including instructions in the prompt for the model to justify its reasoning in its responses[17]. This technique is helpful in prompt refinement and the manual verification of model responses, though it also has some limitations (see Chen et al. 2025).

Include representative examples across categories. Including a few highly representative examples in the prompt is critical for ensuring that the model generates relevant responses, a process known as in-context learning (see Zewe 2023) or multi-shot prompting (see Anthropic n.d.). Aim to use at least five examples, depending on the model's context length (see Bergmann 2024a). For the literature identification task, we included approximately five representative, manually labeled papers in our prompts, with at least one relevant and one irrelevant example.

Include a request for references. Asking the model to include references to the source document(s) in its response can help with prompt refinement. For instance, if you want the model to generate a synthesis of 20 summaries, be clear that it should cite the specific summaries in its response. The references can be in the form of summaries of the key points from the reference text.

Provide "unknown" or "not applicable" as a category. A limitation of closed LLMs such as OpenAI's GPT-4o is that they are configured to always generate some response, however unlikely it might be given the model's training data. This implies that the model may generate speculative results when it encounters insufficient or low-quality instructions or input data. To mitigate this issue, provide an "unknown" or "not

[16] Humans obviously understand broadly how LLMs work since we designed and developed them.

[17] See World Bank 2024, app B.

applicable" option, which allows the model the option to acknowledge when it does not have sufficient information to carry out the given instructions.

Include a desired response format. It is useful to indicate the desired format in which the LLM should structure its responses. For instance, for the information extraction task, we instructed the model to deliver responses strictly in a specific JSON format, which facilitated the transfer of the responses into a table.

Check edge cases. Check the model's responses on infrequent or highly ambiguous cases to understand the limits of a model's performance in such contexts.

Figure 3. Prompt Format for Literature Identification

Source: Independent Evaluation Group.

Evaluating Model Performance

As mentioned in previous sections, a manual review of model responses is necessary when using LLMs in real evaluation use cases. This section offers some points to keep in mind when developing a strategy for evaluating model performance for an experiment or application.

Assess the faithfulness of responses. Regardless of the type of task for which an LLM is being used, the evaluator should review the faithfulness of the model's responses. For example, in text classification, it is useful to assess the model's justifications using this criterion. Lower-than-expected levels of faithfulness can indicate a flaw in the design of the task, such as very long contexts.

Set context-specific thresholds for selected metrics. Set clear thresholds for the respective model evaluation metrics and ensure that all relevant stakeholders agree with these thresholds, as the threshold defines the level of LLM performance that would be considered satisfactory. Refine the prompt or other aspects of the design until such results are achieved.

Use annotation and validation guidelines. To maintain consistency throughout the validation process, reviewers should use annotation guidelines in the form of a shared codebook (see for example Kallos 2023). The codebook should include the instructions that manual reviewers will need for tasks such as labeling observations for classification or assigning values to assessment metrics for summarization or synthesis.

Check intercoder reliability. During the processes of human data annotation or model response validation, despite the use of a detailed codebook or instructions, disagreements can arise between two or more coders. Calculating an intercoder reliability score such as Cohen's Kappa (Cohen, 1960) is one way to "demonstrate the rigor of coding procedures" (Cheung et al. 2021, 1155) and can help evaluators settle on realistic target values for model performance metrics. In our experiments, the subject matter expert to provide labeled data for the text classification task. During prompt validation, the team discussed the cases where the model's labels differed from the expert's to arrive at a common understanding regarding. In future experiments, we aim to capture this iterative process of arriving at "ground-truth" labels more systematically, for example by using and reporting metrics such as Cohen's Kappa.

Use a confusion matrix for text classification. A confusion matrix (see Murel 2024 for practical guidance) is helpful for summarizing the performance of a classification model because it displays key metrics of interest. This matrix can help an evaluator diagnose a model's classification performance by displaying the number of results that are true positives, true negatives, false positives, and false negatives. Use this knowledge during

prompt refinement (see Refining Prompts) based on what matters most for a use case. For example, in the SLR use case, we wanted to ensure a low rate of false negatives and could accept a higher rate of false positives because we wanted to ensure that we did not miss any relevant papers to include in our review.

Refining Prompts

Use validation findings for prompt refinement. If the results on the validation set do not meet your expected or required threshold, analyze the cause of the inaccuracies and use your findings to refine the prompt. For example, you might notice that the model makes some incorrect assumptions, so you need to include instructions to avoid those. You can see what impact your changes to the prompt have on the confusion matrix for the validation set and adjust the prompt accordingly. For text classification, the confusion matrix serves as a critical tool to help the team understand the sources of errors (for example, false positives or false negatives).

Avoid creating convoluted prompts. As experimentation progresses, it is tempting to continually add instructions to prompts to address edge cases and improve performance. However, over time, doing so can lead to overly complex prompts with a patchwork of fixes, making the prompt susceptible to overfitting (Google 2025) the validation set.

Going Forward

In the World Bank and International Fund for Agricultural Development independent evaluation departments, we have embarked on a journey of experimentation with the application of AI in our practice. This journey is primarily about thoughtful risk taking, continuous learning and adaptation, and dialogue between staff with different areas of expertise. Learning to use AI is not a one-time effort but rather a continuous process of questioning, testing, learning, and refining.

In this guidance note, we focused on two fundamental aspects of this journey: (i) defining and adapting our typical evaluation workflows to include LLMs where they fit best, and (ii) building trust through thorough performance testing (that is, adapting typical criteria of rigor to the specificity of LLM usage). Further research, experimentation, and collaboration are needed to standardize and expand on frameworks for assessing the performance of LLMs in evaluation. Collaboration should include sharing experiences and findings from experiments and pilots across organizations and contexts.

Much has already been written on the potential and perils of leveraging LLMs in research and analytical tasks, but it is in the concrete, practical, context-specific

experimentation that we can find out what works, what does not work, and under what circumstances something either works or does not. We are committed to keep exploring and sharing what we find as widely as possible.

Bibliography

Alaofi, Marwah, Negar Arabzadeh, Charles L. A. Clarke, and Mark Sanderson. 2024. "Generative Information Retrieval Evaluation." *arXiv* preprint, April 11. arXiv:2404.08137v3. https://arxiv.org/abs/2404.08137.

Amazon Web Services. n.d.-a. "Prompt Templates and Examples for Amazon Bedrock Text Models." Amazon Bedrock User Guide. Accessed May 5, 2025. https://docs.aws.amazon.com/bedrock/latest/userguide/prompt-templates-and-examples.html.

Amazon Web Services. n.d.-b. "What Is Compute?" Amazon Web Services. https://aws.amazon.com/what-is/compute/.

Anthropic. n.d. "Use Examples (Multishot Prompting) to Guide Claude's Behavior." Prompt Engineering. Accessed May 5, 2025. https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/multishot-prompting.

Anuj, Harsh, Virginia Ziulu, Ariya Hagh, Estelle Raimondo, and Jos Vaessen. 2025. "World Bank IEG Evaluations and the Role of Data Science: Reflections from Recent Experiences." In *Artificial Intelligence and Big Data: Lessons from Evaluations of the Rule of Law and Development*, edited by Frans L. Leeuw and Michael Bamberger, 231–251. Edward Elgar Publishing. https://www.elgaronline.com/edcollchap/book/9781803925677/chapter11.xml.

Arize. 2025. "The Definitive Guide to LLM Evaluation: A Practical Guide to Building and Implementing Evaluation Strategies for AI Applications." Arize AI. https://arize.com/llm-evaluation.

Banerjee, Satanjeev, and Alon Lavie. 2005. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments." In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72. Association for Computational Linguistics. https://aclanthology.org/W05-0909/.

Bergman, Dave. 2024a. "What Is a Context Window?" IBM Think, November 7. https://www.ibm.com/think/topics/context-window.

Bergmann, Dave. 2024b. "What Is Instruction Tuning?" IBM Think, April 5. https://www.ibm.com/think/topics/instruction-tuning.

Brown, Tom B., Benjamin Mann, Nick Ryder et al. 2020. "Language Models Are Few-Shot Learners." *arXiv* preprint, May 28. arXiv:2005.14165v4. https://arxiv.org/abs/2005.14165.

Chang, Yupeng, Xu Wang, Jindong Wang et al. 2024. "A Survey on Evaluation of Large Language Models." *ACM Transactions on Intelligent Systems and Technology* 15 (3): Article 39, 1–45. https://doi.org/10.1145/3641289.

Chen, Yanda, Joe Benton, Ansh Radhakrishnan et al. 2025. "Reasoning Models Don't Always Say What They Think." Anthropic. https://assets.anthropic.com/m/71876fabef0f0ed4/original/reasoning_models_paper.pdf.

Cheung, Diana. 2024. "An Introduction to LLM Evaluation: How to Measure the Quality of LLMs, Prompts, and Outputs." *Codesmith* (blog), May 15. https://www.codesmith.io/blog/an-introduction-to-llm-evaluation-how-to-measure-the-quality-of-llms-prompts-and-outputs.

Cheung, Kason Ka Ching, and Kevin W. H. Tai. 2021. "The Use of Intercoder Reliability in Qualitative Interview Data Analysis in Science Education." *Research in Science & Technological Education* 41 (3): 1155–75. https://doi.org/10.1080/02635143.2021.1993179.

Cohen, Jacob. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*(1), 37-46. https://doi.org/10.1177/001316446002000104.

DAIR.AI. 2025. "Prompt Engineering Guide." https://www.promptingguide.ai/.

Durmus, Esin, He, and Mona Diab. 2020. "FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5055–70. Association for Computational Linguistics. https://aclanthology.org/2020.acl-main.454/.

Fabbri, Alexander R., Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. "SummEval: Re-evaluating Summarization Evaluation." *Transactions of the Association for Computational Linguistics* 9: 391–409. https://doi.org/10.1162/tacl_a_00373.

Gadesha, Vrunda, Vanna Winland, and Eda Kavlakoglu. 2025. "What is chain of thought (CoT) prompting?". *IBM Blog*, April 23. https://www.ibm.com/think/topics/chain-of-thoughts.

Gera, Ariel, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. "Zero-Shot Text Classification with Self-Training." In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 1107–19. Association for Computational Linguistics. https://aclanthology.org/2022.emnlp-main.73/.

Géron, Aurélien. 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. O'Reilly Media. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/.

Glickman, Mark, and Yi Zhang. 2024. "AI and Generative AI for Research Discovery and Summarization." *arXiv* preprint, January 8. arXiv:2401.06795v2. https://arxiv.org/abs/2401.06795.

Goodwin, Michael. 2024. "What is an API (application programming interface)?" *IBM blog*, April 9. https://www.ibm.com/think/topics/api.

Google. 2024. "Overfitting." Machine Learning Concepts. https://developers.google.com/machine-learning/crash-course/overfitting/overfitting.

Google. 2025. "Machine Learning Glossary." Google for Developers. https://developers.google.com/machine-learning/glossary.

Huyen, Chip. 2023. "Multimodality and Large Multimodal Models (LMMs)". *Blog*, October 10. https://huyenchip.com/2023/10/10/multimodal.html.

Kallos, Alecia. 2023. "Creating a Qualitative Codebook." Eval Academy. https://www.evalacademy.com/articles/creating-a-qualitative-codebook.

Kinney, Rodney, et al. 2023. "The Semantic Scholar Open Data Platform." *arXiv* preprint, January 24. arXiv:2301.10140. https://doi.org/10.48550/arXiv.2301.10140

Lewis, Patrick, Ethan Perez, Aleksandra Piktus et al. 2020. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." In *Advances in Neural Information Processing Systems* 33, 9459–9474. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

Lin, Chin-Yew. 2004. "ROUGE: A Package for Automatic Evaluation of Summaries." In *Text Summarization Branches Out, Association for Computational Linguistics*, July 25, Barcelona. https://aclanthology.org/W04-1013/.

Liu, Jun, Prem Timsina, and Omar El-Gayar. 2018. "A comparative analysis of semi-supervised learning: The case of article selection for medical systematic reviews. *Inf Syst Front* 20, 195–207. https://doi.org/10.1007/s10796-016-9724-0

Liu, Nelson F., Kevin Lin, John Hewitt et al. 2024. "Lost in the Middle: How Language Models Use Long Contexts." *Transactions of the Association for Computational Linguistics* 12: 57–173. https://doi.org/10.1162/tacl_a_00638.

Liu, Yang, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. "G-Eval: NLG Evaluation Using GPT-4 with Better Human Alignment." In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2511–22. Association for Computational Linguistics. https://aclanthology.org/2023.emnlp-main.153/.

Martineau, Kim. 2023. "What Is AI Alignment?" *IBM Research* (blog), November 8. https://research.ibm.com/blog/what-is-alignment-ai.

MacQueen, James. 1967. "Some methods for classification and analysis of multivariate observations." In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5, pp. 281-298. University of California press.

McHugh, Mary L. 2012. "Interrater Reliability: The Kappa Statistic." *Biochemia Medica* 22 (3): 276–82. https://pmc.ncbi.nlm.nih.gov/articles/PMC3900052/.

Mucci, Tim. 2024. "What is data leakage in machine learning?". IBM Think, September 30. https://www.ibm.com/think/topics/data-leakage-machine-learning.

Murel, Jacob. 2024. "Create a Confusion Matrix with Python." IBM Developer, March 7. https://developer.ibm.com/tutorials/awb-confusion-matrix-python/

OpenAI. n.d. "Best Practices for Prompt Engineering with the OpenAI API." OpenAI Help Center. Accessed May 4, 2025. https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-the-openai-api.

OpenAI. 2022. "Clustering." OpenAI Cookbook, March 10. https://cookbook.openai.com/examples/clustering.

OpenAI. 2024. *GPT-4o System Card*. OpenAI. https://cdn.openai.com/gpt-4o-system-card.pdf.

Ouyang, Long, Jeff Wu, Xu Jiang et al. 2022. "Training Language Models to Follow Instructions with Human Feedback." In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. "BLEU: A Method for Automatic Evaluation of Machine Translation." In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–18. Association for Computational Linguistics. https://doi.org/10.3115/1073083.1073135.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research* 12 (2011): 2825–2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

Puri, Raul, and Bryan Catanzaro. 2019. "Zero-Shot Text Classification with Generative Language Models." *arXiv* preprint, December 10. arXiv:1912.10165v1. https://arxiv.org/abs/1912.10165.

Raimondo, Estelle, Harsh Anuj, and Virginia Ziulu. 2023a. "Setting up Experiments to Test GPT for Evaluation." *IEG Blog* (blog), August 16. https://ieg.worldbankgroup.org/blog/setting-experiments-test-gpt-evaluation.

Raimondo, Estelle, Virginia Ziulu, and Harsh Anuj. 2023b. "Fulfilled Promises: Using GPT for Analytical Tasks." *IEG Blog* (blog), August 23. https://ieg.worldbankgroup.org/blog/fulfilled-promises-using-gpt-analytical-tasks.

Raimondo, Estelle, Harsh Anuj, and Virginia Ziulu. 2023c. "Unfulfilled Promises: Using GPT for Synthetic Tasks." *IEG Blog* (blog), August 30. https://ieg.worldbankgroup.org/blog/unfulfilled-promises-using-gpt-synthetic-tasks.

Shin, Taylor, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, edited by Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, 4222–35. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.346.

van der Lee, Chris, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. "Best Practices for the Human Evaluation of Automatically Generated Text." In *Proceedings of the 12th International Conference on Natural Language Generation*, 355–68. Association for Computational Linguistics. https://aclanthology.org/W19-8643/.

Wang, Zhiqiang, Yiran Pang, and Yanbin Lin. 2023. "Large Language Models Are Zero-Shot Text Classifiers." *arXiv* preprint, December 2. arXiv:2312.01044v1. https://arxiv.org/abs/2312.01044.

World Bank. 2017. *Conducting a Structured Literature Review in the Framework of IEG (Major) Evaluations*. IEG Methods Literature. Independent Evaluation Group. World Bank.

World Bank. 2024. *Biodiversity for a Livable Planet: An Evaluation of World Bank Group Support for Biodiversity (FY15–24)*. Approach Paper. Independent Evaluation Group. World Bank. https://ieg.worldbankgroup.org/sites/default/files/Data/reports/ap_biodiversity.pdf.

World Bank. Forthcoming. *Epidemic Preparedness*. Approach Paper. Independent Evaluation Group. World Bank.

Yan, Ziyou. 2024. "Task-Specific LLM Evals That Do and Don't Work." eugeneyan.com. https://eugeneyan.com/writing/evals/.

Zewe, Adam. 2023. "Solving a Machine-Learning Mystery. MIT News, February 7. https://news.mit.edu/2023/large-language-models-in-context-learning-0207.

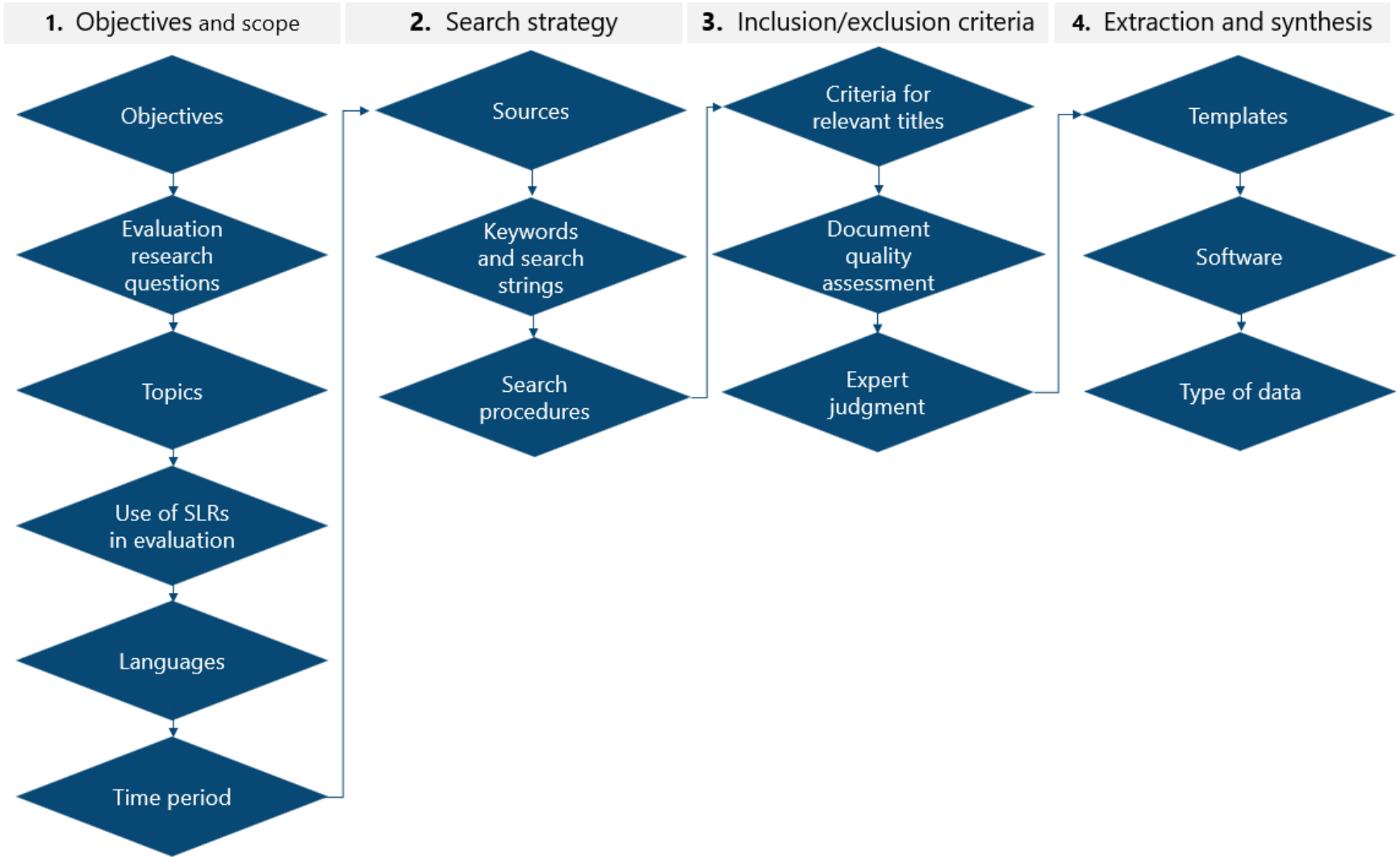
Zhang, Tianyi, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. "Benchmarking Large Language Models for News Summarization." *Transactions of the Association for Computational Linguistics* 12: 39–57. https://doi.org/10.1162/tacl_a_00632.

Ziulu, Virginia, Harsh Anuj, Ariya Hagh, Estelle Raimondo, and Jos Vaessen. 2024. "Extracting Meaning from Textual Data for Evaluation: Lessons from Recent Practice at the Independent Evaluation Group of the World Bank." In *Artificial Intelligence and Evaluation: Emerging Technologies and Their Implications for Evaluation*, edited by Steffen Bohni Nielsen, Francesco Mazzeo Rinaldi, and Gustav Jakob Petersson, 57–73. Routledge. https://www.taylorfrancis.com/chapters/oa-edit/10.4324/9781003512493-5/extracting-meaning-textual-data-evaluation-virginia-ziulu-harsh-anuj-ariya-hagh-estelle-raimondo-jos-vaessen.

Appendix A. Additional Workflows

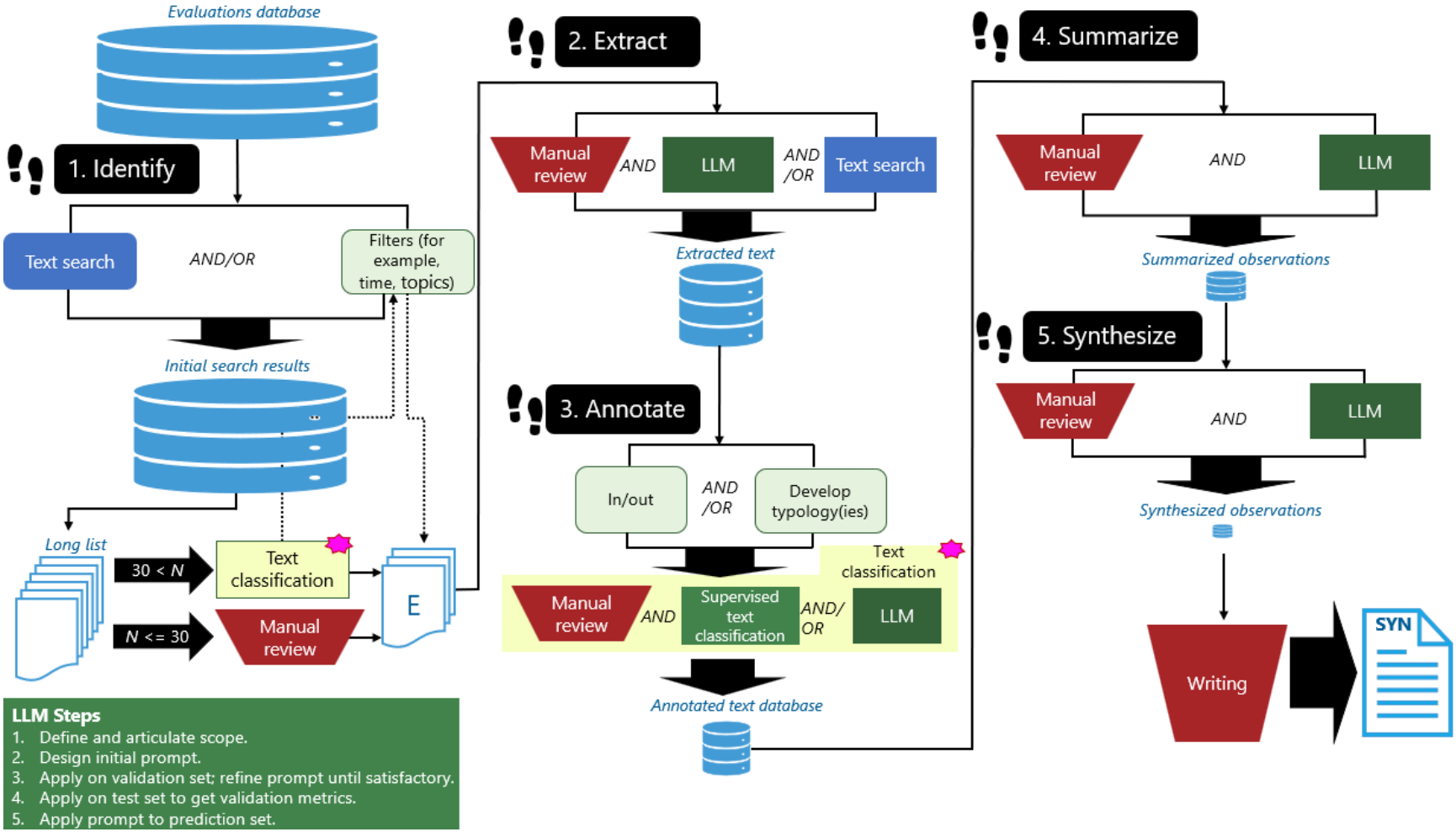
Figure A.1 provides an alternative workflow for structured literature reviews (SLRs) in the framework of Independent Evaluation Group major evaluations. This workflow is based on a checklist for conducting such reviews provided as internal methodological guidance and is closer to the "traditional" approach. We provide this as a comparison to the workflow presented in figure 1 to demonstrate the slightly different framing when viewing the same use case from the perspective of different specializations or domains. We hope that such a comparison will help evaluators think through how they can translate their workflows from ones similar to figure A.1 to ones more like figure 1 to enable the application of large language models. Figure A.2 provides our current proposed workflow for evaluation synthesis. Indeed the same workflow can be replicated across use cases, including portfolio review and analysis and interview transcripts analysis. IEG is currently piloting the former as a set of AI-powered web-based applications developed in-house jointly with the WB Information Ttechnology Solutions department (ITS).

Figure A.1. Alternate Workflow for Structured Literature Reviews



Source: World Bank 2017.

Figure A.2. Evaluation Synthesis Workflow



Source: Independent Evaluation Group.
Notes: LLM = Large Language Model.

Reference

World Bank. 2017. *Conducting a Structured Literature Review in the Framework of IEG (Major) Evaluations*. IEG Methods Literature. Independent Evaluation Group. World Bank.