METADATA-GUIDED GENERATION OF DOMAIN-SPECIFIC PEER REVIEWS WITH LLMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

EZGI ÇAVUŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

AUGUST 2025

Metadata-Guided Generation of Domain-Specific Peer Reviews with LLMs

submitted by **EZGI ÇAVUŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç
Dean, **Graduate School of Informatics** _____

Prof. Dr. Altan Koçyiğit
Head of Department, **Information Systems** _____

Prof. Dr. Tuğba Taşkaya Temizel
Supervisor, **Data Informatics, METU** _____

Examining Committee Members:

Prof. Dr. Altan Koçyiğit
Information Systems, METU _____

Prof. Dr. Tuğba Taşkaya Temizel
Data Informatics, METU _____

Assoc. Prof. Dr. Nurcan Alkış Bayhan
Computer Engineering, Baskent University _____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ezgi Çavuş

Signature :

ABSTRACT

METADATA-GUIDED GENERATION OF DOMAIN-SPECIFIC PEER REVIEWS WITH LLMS

Çavuş, Ezgi

M.S., Department of Information Systems

Supervisor: Prof. Dr. Tuğba Taşkaya Temizel

August 2025, 71 pages

Large Language Models (LLMs) are increasingly used to support academic evaluation, yet they often struggle to capture the domain-specific criteria essential for rigorous peer review. While existing reviewer guidelines and research checklists address general concerns such as reproducibility, ethical considerations, acknowledgment of limitations, and societal impact, they often neglect discipline-specific criteria that vary with a paper's methodology, or evaluation strategy. This thesis introduces a framework that automatically extracts domain-specific review questions from past evaluations in the OpenReview system and aligns them with new submissions using structured metadata derived from paper content, including methodology, datasets, and evaluation metrics. The results show that the proposed framework improves the specificity and contextual relevance of generated reviews. Compared to baseline models, it allows more precise control over the content and focus of generated reviews by explicitly grounding them in metadata and past evaluation patterns, even when overall performance metrics are similar. The system improves traceability, reduces hallucinated content, and increases the interpretability of automated peer review.

Keywords: domain specific peer review, natural language processing, question generation, metadata extraction

ÖZ

ALANA ÖZGÜ HAKEM YORUMLARININ BÜYÜK DİL MODELLERİ İLE ÜSTVERİ REHBERLİĞİNDE ÜRETİLMESİ

Çavuş, Ezgi

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi: Prof. Dr. Tuğba Taşkaya Temizel

Ağustos 2025, 71 sayfa

Büyük Dil Modelleri (LLM'ler) akademik değerlendirmeyi desteklemek için giderek daha fazla kullanılmaktadır, ancak genellikle titiz bir hakemlik süreci için gerekli olan alan-özel ölçütleri yakalamakta zorlanırlar. Mevcut hakem yönergeleri ve araştırma kontrol listeleri, çoğaltılabilirlik, etik değerlendirmeler, sınırlamaların belirtilmesi ve toplumsal etki gibi genel konulara değinse de, genellikle bir makalenin metodolojisine, veri kümesine veya değerlendirme stratejisine göre değişen disipline özgü ölçütleri göz ardı ederler. Bu tez, OpenReview sistemindeki önceki değerlendirmelerden alan-özel inceleme sorularını otomatik olarak çıkaran ve bunları metodoloji, veri kümesi ve değerlendirme metrikleri gibi makale içeriğinden türetilmiş yapılandırılmış üstveriler aracılığıyla yeni başvurularla hizalayan bir çerçeve sunar. Sonuçlar, önerilen çerçevenin üretilen incelemelerin özgüllüğünü ve bağlamsal uygunluğunu artırdığını göstermektedir. Temel modellerle karşılaştırıldığında, genel performans ölçütleri benzer olsa da incelemelerin içeriği ve odağı üzerinde daha hassas kontrol sağlar; çünkü incelemeleri açık bir şekilde üstveri ve geçmiş değerlendirme örüntüleriyle temellendirir. Sistem, izlenebilirliği artırır, uydurulmuş içerikleri azaltır ve otomatikleştirilmiş hakemli değerlendirmenin yorumlanabilirliğini artırır.

Anahtar Kelimeler: alan odaklı hakem değerlendirmesi, doğal dil işleme, soru üretimi, meta veri çıkarma

To my family

ACKNOWLEDGMENTS

First and foremost, I extend my profound gratitude to Prof. Dr. Tuğba Taşkaya Temizel for her unwavering support, patience, and invaluable insights throughout the course of this thesis. Her guidance ensured that I remained focused, steering me in the right direction and ultimately making this work possible.

I would like to thank my family for their constant support.

Besides my family and supervisor, I would like to thank my colleagues for their help, encouragement, and friendship during my thesis process and master's study.

I thank the Turkish Scientific and Technological Research Council (TUBITAK) for their financial support with the Directorate of Science Fellowships and Grant Programmes (BIDEB) 2210-A program during my study.

I am grateful for the computational resource support provided by the Deep Learning and Computer Vision Research Lab, METU DXTE, and AI & Big Data Lab for letting me go over extensive experiments.

This journey would not have been possible without the support of my husband and best friend, Barış Çavuş. I am so grateful to him for his companionship and for being my constant encouragement every step of the way.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ v

DEDICATION vi

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES xiii

LIST OF FIGURES xiv

LIST OF ABBREVIATIONS xv

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Scope of the Study 2

1.3 Research Questions 3

1.4 The Outline 4

2 RELATED WORK 5

2.1 Automated Scientific Review 5

2.2 Review Generation Application Studies 6

2.3 Question Generation 7
2.4 Prompt Engineering 10
2.5 LLM for Annotation 11
3 METHOD 13
3.1 Dataset and Exploratory Data Analysis 14
3.1.1 Paper and Review Data 14
3.1.2 Aspect-Based Annotated Data 15
3.1.3 Statistical Analysis of Aspect–Acceptance Relationship 16
3.1.4 Preprocessing 21
3.2 Prompt Construction 21
3.2.1 Pilot Study 22
3.2.2 Prompt Construction For Metadata Generation 22
3.2.3 Prompt-Response Workflow for Question Extraction, Generalization, and Post Processing 23
3.3 Metadata-level Generalization of Review Questions 25
3.3.1 Entity List Creation 25
3.3.2 Iterative Generalization and Marking 26
3.4 Matching Reviewer Questions via Metadata Embeddings 28
Embedding and Indexing. 28
Similarity-Based Matching. 28
Coverage Aggregation Across Combinations. 28
Representative Question Selection. 29
4 EXPERIMENTS 31

4.1 Experimental Setup 31
4.1.1 Computational Resources 31
4.1.2 Initial Selection of Baseline model 31
4.2 Results and Evaluation 32
4.2.1 Effectiveness of the Proposed Metadata-Guided Question Matching . . . 33
4.2.2 Question Matching Evaluation Against Extracted Review Questions . . . 33
4.2.3 Review Generation and Evaluation Against Ground Truth Reviews 34
Model Configurations. 34
Evaluation Criteria. 34
4.2.4 Review Generation Evaluation 35
Review-Level Similarity Results. 35
Limitations of Review-Level Evaluation. 35
Sentence-Level Evaluation Methodology. 36
Sentence-Level Similarity Results. 36
Aspect-Based Filtering. 36
Interpretation of Sentence-Level Trends. 37
4.3 Qualitative Study 38
4.3.1 Data Preprocessing 38
4.3.2 Metadata Extraction 39
4.3.3 Qualitative Comparison of Golden and Matched Questions 39
4.3.4 Question Set Diversification and Comparison 41
4.3.5 Reviewer Guidelines from Major Conferences vs Proposed Framework 41

4.3.6 Comparison of Ground Truth and Generated Reviews 44

4.4 Discussion 49

5 CONCLUSION 53

5.1 Conclusion 53

5.2 Limitations 54

5.3 Future Work 55

REFERENCES 57

APPENDICES

A EXTRA MATERIAL 61

A.1 Practical Guideline for Applying the Methodology 61

A.2 Prompt Trials 63

A.2.1 Primary Question Extraction Prompt Example 63

A.2.2 Final Question Generation 64

A.2.3 Final Question Generalization 64

A.2.4 Question Post Processing Chain 65

A.2.5 Evaluation Metric Text Extraction from Paper Content 65

A.2.6 Evaluation Metric Extraction 65

A.2.7 Dataset Text Extraction from Paper Content 66

A.2.8 Dataset Extraction 66

A.2.9 Data Type Extraction 66

A.2.10 Training Strategy Text Extraction from Paper Content 66

A.2.11 Training Strategy Extraction 67

A.2.12 Model Text Extraction from Paper Content 67

A.2.13 Model Type Extraction 67

A.3 July 2025 IEEE Taxonomy 68

A.4 Final Entity List 69

A.5 Pilot Study 70

LIST OF TABLES

Table 1 Basic statistics of ASAP-Review dataset for ICLR proceedings. 15

Table 2 Combined statistical results for each aspect, sorted by logistic regression coefficient. Significant results ($p < 0.05$) are in bold. 17

Table 3 Prompt Trials for Metadata Extraction 23

Table 4 Prompt Trials for Reviewer Question Extraction Steps 24

Table 5 BERTScore, embedding similarity, and execution time for models on review generation without reviewer questions on 100 papers 32

Table 6 Mean coverage values of the best covering pairwise metadata feature combinations. 33

Table 7 Review-wise Similarity Scores Across Model Variants 35

Table 8 Sentence-wise Similarity Scores Across Model Variants 36

Table 9 Sentence-wise Similarity Scores After Filtering Aspects Across Model Variants 37

Table 10 Examples of sentence pairs aligned by the Hungarian algorithm for qwen2.5_mq_v2 alongside their matched BERT scores 37

Table 11 Qualitative comparison of overlapping golden and matched questions 40

Table 12 Number of NA values for each metadata among 79 non-theorethical papers . . 40

Table 13 Matched new proposed framework questions with guideline questions and thematic aspects 42

Table 14 Review Comparison for ICLR_2020_395 45

LIST OF FIGURES

Figure 1 Two main phases of proposed methodology. 13

Figure 2 Feature importance and aspect coefficients for acceptance/rejection decisions by year. 18

Figure 3 Aspect coverage distribution for each year and overall total. 20

Figure 4 Workflow for LLM-assisted Generation of a Structured Questions and Entity List. 27

Figure 5 Matching unseen papers with prior questions using FAISS indexes built on metadata combinations 29

Figure 6 Computation pipeline for semantic coverage ratio using matched questions. 29

Figure 7 Representative question selection pipeline using clustering over high-coverage question combinations. 30

LIST OF ABBREVIATIONS

ACL	Association for Computational Linguistics
ASAP-Review	ASpect-enhAnced Peer Review
ASPR	Automated Scholarly Paper Review
Auto-PRE	Automatic and cost-efficient peer-review framework
BART	Bidirectional and Auto-Regressive Transformers
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual evaluation understudy
CIFAR	Canadian Institute For Advanced Research
FAISS	Facebook AI Similarity Search
GAT	Graph Attention Network
GPT	Generative Pre-Trained Transformers
ICLR	International Conference on Learning Representations
KG	Knowledge Graph
LLM	Large Language Model
LLaMa	Large Language Model Meta AI
Longformer	The Long-Document Transformer
LSTM	Long Short Term Memory
MAE	Mean absolute error
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MLP	Multi-Layer Perceptron
MOPRD	Multidisciplinary Open Peer-Review Dataset
NeurIPS	Conference on Neural Information Processing Systems

NLP	Natural Language Processing
QG	Question Generation
QA	Question answering
QuALITY	Question answering with long input texts, yes!
RACE	ReAding Comprehension Dataset
RAG	Retrieval-Augmented Generation
RE	Requirements Engineering
RMSE	Root mean-square error
RoBERTa	Robustly Optimized BERT Pretraining Approach
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SQuAD	Stanford Question Answering Dataset
SWAG	Situations With Adversarial Generations
T5	Text-To-Text Transfer Transformer
T-CQG	Topic-Controlled Question Generation
XAI	Explainable AI

CHAPTER 1

INTRODUCTION

1.1 Motivation

The accelerated development of scientific research and technology has led to an enormous growth in peer-reviewed publications. As the volume and complexity of submissions increase, ensuring high-quality, fair, and domain-relevant peer reviews has become a growing challenge. Traditional peer review is widely criticized for being slow, inconsistent, and resource-intensive. Articles may take months or even years to be reviewed, and inconsistent reviewer judgments often result in unreliable or unfair outcomes. Moreover, the growing volume of submissions places an unsustainable burden on the academic community.

Recent efforts have explored the use of large language models (LLMs) to automate the review process and LLMs offer scalability and fluency in assisting reviewers or even generating full reviews [1][2]. Another approach is to directly integrate LLMs into review generation without incorporating domain-specific criteria [3]. All of these methods often overlook the detailed expectations of expert reviewers across different scientific sub-fields, resulting in reviews that are either too shallow or misaligned with disciplinary norms. Thus, current LLM-based review assistants often fail to reflect the specificity, depth, and domain sensitivity of expert-written reviews. Most rely on generic review templates or prompt designs that are not based on crucial points that differ between research areas. A core shortcoming of generic systems is their inability to utilize domain-relevant reviewer questions, the implicit yet critical questions reviewers ask when evaluating papers in specialized subfields. For instance, the evaluation criteria for a theoretical contribution in optimization differ significantly from those for an empirical benchmark paper in natural language processing. Although existing approaches often rely on predefined review templates or evaluation checklists provided by conferences and journals [2], this work takes a different route. Rather than reusing generic review guidelines, reviewer questions are extracted directly from past peer reviews. This allows for the identification of domain-specific evaluative concerns that emerge organically in real reviewing practices, rather than being imposed top-down. The extracted questions reflect actual reviewer expectations across diverse methodologies, data types, and evaluation setups. The motivation for this study is grounded in the need for a more principled and data-driven approach to automated peer review, one that moves beyond templates toward context-aware, reusable, and explainable evaluation mechanisms.

This study directly addresses the following research problem: **How can we use LLMs and structured metadata to automatically generate domain specific questions from past reviews for evaluating new submissions?**

This research contributes to the broader goal of making peer review more efficient, interpretable, and adaptive to the methodological diversity of scientific research.

1.2 Scope of the Study

LLMs such as ChatGPT are widely used to help write academic reviews. However, these models often generate generic and general-purpose feedback, lacking alignment with domain-specific reviewer expectations. This study addresses the challenge of making the peer review process more transparent and useful—both for reviewers and for authors—by surfacing the types of questions reviewers implicitly consider when evaluating papers in specific domains.

The primary problem explored in this work concerns how reviewer questions can be generated, adapted, and aligned for domain-relevant review generation, and how structured, paper-specific metadata can be defined and utilized to guide this process. The following challenges are investigated:

- How reviewer questions can be extracted from existing review texts,
- How these questions can be generalized to fit new submissions across varying methodological or topical contexts,
- How they can be aligned through structured metadata,
- How metadata aligned questioning affect LLM-based domain-specific peer reviews.

As a starting point, reviewer comments from a diverse set of papers are analyzed to identify implicit evaluation questions. These questions are then associated with structured paper features such as model name, data type, dataset, training strategy, and evaluation metric. This association forms the basis for a metadata-guided pipeline designed to support question generation for future submissions.

Metadata plays a critical role in aligning the extracted reviewer questions with new submissions in a principled and scientifically grounded way. Encoding key domain-relevant information allows the system to determine whether a question extracted from one paper is applicable to another. These metadata features enable the creation of diverse and context-appropriate subsets of reviewer questions through meaningful combinations. In this framework, this metadata-guided matching forms the backbone of connecting old and new submissions and ensures that retrieved questions are aligned with the domain and methodology of each paper.

The focus was on extracting more targeted and domain-specific reviewer questions by analyzing actual reviewer comments across a diverse set of papers, instead of relying solely on general-purpose review templates (e.g., the ICLR(International Conference on Learning Representations) review form). For example, a generic question such as “Are claims in the

abstract/introduction accurate and reflective of contributions?" may be replaced with a more targeted one related to the study domain, such as "Do the experimental results adequately demonstrate the contribution of each component in the proposed approach?", which better captures experiment-related or domain-specific considerations. Reviewer criteria vary significantly depending on methodological choices, data types, or comparison baselines.

The main goal of this study is to develop an explainable and reusable framework that can extract domain-specific questions from past reviews and align them with new submissions based on paper-specific metadata.

1.3 Research Questions

This research is guided by the following questions:

- **RQ1:** How can reviewer questions be systematically generated to support the evaluation of scientific papers?
  - What methods are effective in extracting meaningful reviewer questions from existing review texts?
  - How can extracted questions be generalized across different domains and paper types?
- **RQ2:** How can reviewer questions be adapted and aligned to new submissions?
  - What forms of paper metadata (e.g., method, dataset, task) are most relevant for aligning questions?
  - How can it be ensured that aligned questions remain contextually appropriate and evaluative?
- **RQ3:** How should metadata be defined and structured to enable effective question-paper matching?
  - What types of entities and attributes best capture the dimensions that reviewer expectations depend on?
  - How can paper metadata be represented to support automated matching and prompting?
- **RQ4:** What is the impact of incorporating retrieved and aligned reviewer questions into LLM-based domain-specific review generation?
  - How does question-guided prompting affect the quality, relevance, and traceability of generated reviews?
  - Can this approach lead to more domain-sensitive and interpretable review outputs?

To address these questions, the study introduces a structured pipeline involving review question extraction, metadata-based matching, and question-driven prompting. The scope includes

prompt engineering, entity-based generalization, semantic matching via embedding models, and both qualitative and quantitative evaluation of review outputs.

The scope of the study is limited to submissions to the ICLR conference, including both accepted and rejected papers. For the pilot study and quantitative analysis of reviewer questions and metadata, ICLR submissions from 2017 to 2020 were used. For the qualitative analysis, a representative subset of ICLR 2025 submissions was analyzed to evaluate alignment and applicability of the extracted questions.

1.4 The Outline

The organization of this thesis is as follows: Chapter 2 states the relevant literature on automated review generation via LLMs, question extraction, and prompt engineering, and use of LLMs in annotation of text, establishing the conceptual basis for the research. Chapter 3 outlines the explanatory data analysis, methodology used for prompt-based extraction, detailing how various prompt formats and strategies were iteratively developed and refined to extract both metadata and reviewer questions. It also states how question matching based on metadata of papers is done. Chapter 4 presents the experimental framework, including experimental setup, metadata-guided matching evaluations, and discusses the outcomes through both quantitative metrics and qualitative analysis. Chapter 5 interprets the conclusion in a broader context, reflecting on the potential of LLMs for supporting peer review and suggesting future directions for improvement.

CHAPTER 2

RELATED WORK

This section covers key areas relevant to this study. Automated scientific review methods aim to improve peer review automation. Question generation focuses on creating targeted reviewer queries. Prompt engineering involves testing different prompts to get the best results from language models. LLMs are used for annotation to efficiently generate metadata by labeling specific paper sections. These areas form the basis of this approach, which combines targeted question extraction and optimized metadata generation to enhance automated scientific reviews.

2.1 Automated Scientific Review

The increasing number of journal submissions pressures the peer review process. In response to this challenge, proposals have been made to streamline the initial reviewing of submissions, aiming to save considerable working hours and potentially improve overall academic productivity. Several platforms[4][5][6] have already adopted automated screening tools to identify plagiarism and check formatting requirements. Some of these tools go further by assessing the quality of a study or providing summarizations, thus decreasing the workload on reviewers.

Recent advances in artificial intelligence (AI) have opened possibilities for (semi-) automated peer review systems [7]. These systems could play a role in flagging potentially controversial studies and automating the matching of reviewers with specific manuscripts. However, ethical concerns have arisen, particularly regarding biases and the potential replication of biases by AI systems.

One of the related studies [8] discusses technology's role in automating various aspects of peer review processes in publishing, including suggesting journals, quality control, reviewer selection, and review evaluation. It explores attempts to assess article quality from peer review text and scores but not from bibliometric data. Automation is found beneficial for finding reviewers and assisting with initial quality control, but academic evaluations of other publishing software are lacking. The effectiveness of AI in supporting peer review remains unclear. Additionally, while peer review text and scores could be valuable for post-publication research assessment, their availability for systematic automation is currently limited.

A study by Basuki et al. [9] developed a prediction model for paper quality assessment in peer review, which helped to reduce the burden on reviewers. The model relies on paper features

rather than reviewer comments and addresses multiple tasks, including final decision prediction, paper quality estimation, and review score prediction. Key aspects of their approach include modeling citation functions that represent the intentions of the authors behind the citations, such as introducing topics, comparing, criticizing, or supporting prior work, which are used as functional cues in the text to predict future impact of the article. Their framework employs several predictors: a citing sentence predictor analyzing citation-containing sentences; a regular sentence predictor for non-citation sentences classified into semantic categories via a BERT(Bidirectional Encoder Representations from Transformers) model; and a reference-based predictor that incorporates metadata and features from reference sections. Using the ASAP-Review(Aspect Enhanced Peer Review) dataset as utilized in this thesis, the model performs both classification (high, medium, low citation groups) and regression (predicting exact citation counts). The classification accuracy reaches up to 98.33% , and regression metrics such as RMSE(Root mean-square error) and MAE(Mean absolute error) are around 0.3 for highly cited papers, demonstrating strong predictive performance. Citation function-based predictors—especially the citing sentence predictor—were highly effective, with the "citing paper dominant" feature playing a significant role in improving prediction quality.

Peer review is important for evaluating research in academic publishing, yet it faces criticism for its inefficiencies and low reproducibility. Recent years have witnessed AI integration to assist in peer review, but limitations persist due to human involvement. One of the studies [10] proposes the concept and pipeline of Automated Scholarly Paper Review (ASPR) and reviews literature and technologies for achieving a comprehensive computerized review process. The analysis concludes that corresponding research and preliminary implementations exist at each stage of ASPR. Challenges in ASPR with current technologies include inadequate data, imperfect document parsing, inadequate human-computer interaction, and flawed deep logical reasoning. The paper also discusses future directions and considers the moral and ethical implications of ASPR. It anticipates a coexistence of ASPR and traditional peer review, reinforcing each other until ASPR can independently handle the reviewing workload.

Lin et al.[11] highlight the growing adoption of open peer review in academic publishing and its benefits for the research community. They note that available peer review data enable research on review generation and automated scholarly reviews but suffer from limited scope and a focus mainly on computer science. To overcome these issues, they introduce the Multidisciplinary Open Peer-Review Dataset (MOPRD), which includes paper metadata, multiple manuscript versions, review comments, meta-reviews, author rebuttals, and editorial decisions. They also propose a modular guided review comment generation method using MOPRD, demonstrating improved performance in both automatic metrics and human evaluations.

2.2 Review Generation Application Studies

This section reviews studies on automated review generation applications, selected based on their approaches to simulating peer review, generating constructive comments, and modeling review dynamics. I focused on works that improve LLMs for explainable review processes across diverse tasks, including comment generation, score prediction, and social bias analysis in peer review.

Wang et al. [12] initially developed a comment generation process that selected constructive human review sentences, generalized them into category-specific templates, and converted structured evidence into natural language comments. Building on this, they introduced ReviewRobot [3], an automatic review assistant that generates explainable comments with review scores beside of knowledge graphs(KGs) from domain-specific information on related work section and rest of the paper. Differences between these KGs are analyzed to extract evidence for each review category such as Recommendation, Summary, Meaningful Comparison, Novelty, Soundness, etc. The study was evaluated on 50 ACL2017 papers by domain experts who found 70.5%, 44.6%, and 41.7% of comments constructive and valid for Summary, Novelty, and Meaningful Comparison, respectively, with high inter-annotator agreement. Yet, the study is stated as lack of understanding experiment related details of papers in comment generation task. Following the efforts to simulate the peer-review process through human-aligned review generation, more recent approaches extend this idea toward fully automated, scalable, and cost-efficient evaluation of generative models. Chen et al. [13] propose Auto-PRE(Automatic and cost-efficient peer-review framework), an automatic and cost-efficient peer-review framework for language generation evaluation. Unlike prior methods that still rely on human-authored content as reference points or heavily preprocessed real reviews, Auto-PRE proposes a fully automated framework that evaluates LLMs by assigning other LLMs as reviewers, selected based on qualities like consistency, confidence, and task relevance. A qualification exam is employed to ensure only capable models act as evaluators, eliminating the need for human references altogether. The framework has been shown to match or surpass state-of-the-art evaluation quality across diverse generation tasks, including summarization and dialogue, while significantly reducing evaluation costs and mitigating single-model bias. By operationalizing the peer-review analogy in a reference-free and task-agnostic fashion, Auto-PRE represents a scalable and unbiased alternative for LLM evaluation. In peer review generation task, Jin et al.[14] studied the agent-based review generation and proposed an LLM-based simulation framework called AGENTREVIEW: Exploring Peer Review Dynamics with LLM Agents. It shifts the focus from static review generation to dynamic system-level analysis of peer review processes. AGENTREVIEW simulates the complete peer review pipeline using LLM agents acting as reviewers. This allows researchers to explore how latent sociological factors, such as social influence, altruism fatigue, and authority bias, affect review outcomes in the aggregate. Through large-scale simulations involving over 10,000 reviews and 500+ paper submissions, the framework reveals that reviewer bias alone accounts for approximately 37.1% of variation in acceptance decisions. Importantly, AGENTREVIEW addresses privacy concerns by eliminating the need for real reviewer data and offers an open-source, extensible platform for future research.

2.3 Question Generation

Question answering (QA) systems are NLP tools that automatically respond to user questions posed in natural language with correct answers[15]. Common QA task types in the literature includes: ***Extractive (reading comprehension) QA***[16], ***Natural Language Inference (textual entailment)***[17], ***Claim verification (fact checking)***[18],***Multi-hop reasoning QA***[19], ***Knowledge-base (KB) QA***[20]. These tasks and datasets define the landscape of QA research, covering everything from simple fact lookup to complex reasoning and answer verification.

In question generation research, standard benchmark datasets such as SQuAD(Stanford Question Answering Dataset)[16], RACE(ReAding Comprehension Dataset)[21], SWAG(Situations With Adversarial Generations)[22], NewsQA[23] have been widely utilized for various domains and tasks to train and evaluate models [24]. These datasets include human-authored questions, which serve as a reference for assessing the quality of machine-generated questions. Evaluation typically involves two main dimensions. First, the semantic alignment between the generated question-answer pair and the corresponding human-written pair is measured, often through metrics assessing answer coverage and relevance. Second, statistical measures such as diversity ratio (the proportion of unique questions) and compression ratio (the level of redundancy reduction) [25] are applied to quantify the lexical and semantic variety within the generated question set. A higher diversity ratio is generally interpreted as an indicator of richer and more informative question generation.

The work by Zhang et al.[26] offers an overview of the field of QG(Question Generation) from natural language input. It outlines a structured taxonomy that classifies QG tasks based on the granularity of the source text (ranging from documents to keywords), the availability of the target answer (distinguishing between answer-aware and answer-agnostic settings), and the form of the resulting questions. The review contrasts early rule-based techniques with more recent neural network-based methods, particularly focusing on enhanced sequence-to-sequence (Seq2Seq) models that use features like answer location to improve question quality. In addition to methodology, the paper explores practical use cases of QG such as augmenting datasets for QA systems, supporting educational tools, and powering conversational agents. It also discusses evaluation frameworks, including both automatic metrics (such as BLEU(Bilingual evaluation understudy), METEOR(Metric for Evaluation of Translation with Explicit ORdering), and ROUGE(Recall-Oriented Understudy for Gisting Evaluation)) and human-based assessments related to fluency, relevance, and effectiveness. As such, the paper provides a valuable foundation for understanding the core principles and recent advancements in generating questions from text.

A study by Zhang et al.[27] explores the question generation capabilities of LLMs including GPT-3.5(Generative Pre-Trained Transformers), GPT-4, and LLaMA 2-Chat 13B(Large Language Model Meta AI). The findings show that these models are capable of generating high-quality, contextually grounded questions across various domains, particularly in educational and evaluation settings. The study employs multi-step pipelines, first identifying answer spans before generating questions, to further improve specificity and focus, then generating questions with abstractive answer and answer span. They produce questions that align well with the context provided and the target answers, often achieving strong expert ratings in terms of clarity, relevance, and cognitive depth. Advanced prompting strategies, such as chain-of-thought reasoning or taxonomy-guided generation, enable LLMs to formulate questions across different levels of Bloom's taxonomy [28], a hierarchy of cognitive skills from basic recall to higher-order analysis and evaluative reasoning used for extracting abstractive answers. In the result of the study, LLM-generated questions tend to exhibit broader contextual coverage compared to human-authored questions, mitigating positional bias and improving answer completeness , and unanswerable or overly simplistic questions still arise. Consequently, effective prompt engineering and human-in-the-loop validation remain critical to maintaining question quality. These capabilities make LLMs a promising tool for robust question generation in educational technologies and evaluation-driven NLP pipelines.

Several studies have focused on generating questions from input text. One notable example is the work of Du et al. [29], which explores the generation of questions based on input encoded at the paragraph and sentence level. Their approach uses an attention-based sequence learning model that aims to mimic how humans generate questions from a given passage. The model is trained to identify key words or phrases in the input text and transform them into relevant questions. The results indicate that questions generated from sentence-level encodings tend to be more similar to human-generated questions compared to those generated from paragraph-level inputs. This is attributed to the fact that paragraph-level input introduces additional noise, making it harder for the model to focus on salient information.

The QuALITY(Question answering with long input texts, yes!) dataset [30] introduces a challenging multiple-choice question answering benchmark specifically designed to evaluate model performance on long-context understanding. With English passages, the dataset significantly exceeds the input length of typical QA corpora. Questions are crafted by annotators who fully read each passage, ensuring that answering requires integrating information across the entire document rather than relying on extractive or shallow cues. Evaluation results show that current state-of-the-art models, including Longformer (The Long-Document Transformer), RoBERTa(Robustly Optimized BERT Pretraining Approach), and T5(Text-To-Text Transfer Transformer), perform well below human baselines—underscoring the complexity of the task. Additionally, the dataset offers a "HARD" subset to further benchmark deep reasoning and comprehension. QuALITY thus serves as a valuable resource for advancing question generation and answering over extended and information-rich texts.

DataMorgana [25] is a recent framework designed to generate diverse, domain-specific QA datasets for evaluating Retrieval-Augmented Generation (RAG) systems. It uses a semi-structured configuration file specifying question types and user categories to construct prompts for LLMs. Through a two-stage process—prompt generation and candidate filtering—it produces QA pairs that are contextually relevant, category-aligned, and faithful to source documents. Unlike traditional approaches that rely on fixed prompts or leave diversity control entirely to the model, DataMorgana explicitly optimizes for lexical, syntactic, and semantic diversity.

In the context of question generation in educational domain, Li et al.[31] introduce a novel approach called Topic-Controlled Question Generation (T-CQG), which focuses on generating questions that are both semantically coherent and topically aligned with specific educational objectives. The method fine-tunes a lightweight T5-small model using carefully curated datasets, offering a scalable and resource-efficient alternative to larger LLMs. Unlike generic question generation systems, T-CQG incorporates a target topic alongside paragraph-level input to ensure that the resulting questions remain focused and pedagogically relevant. The approach also explores improvements via domain-adaptive pretraining, quantisation, and data augmentation, which improve both performance and scalability. Comprehensive evaluations, including novel automatic metrics and human assessments, demonstrate the effectiveness of the method in producing high-quality, topic-relevant questions suitable for personalized tutoring systems.

2.4 Prompt Engineering

Prompts guide LLMs to produce structured outputs, while prompt engineering shapes model behavior and improves task-specific performance. To support effective prompt design, several studies have examined common patterns and design principles.

White et al.[32] propose a catalog comprising a structured set of reusable patterns aimed at automating processes, generating desired outputs, and combining multiple prompts. Their catalog includes patterns such as role assignment, chain-of-thought prompting, and iterative refinement, along with guidelines for when and how to use them. The framework of documenting patterns serves as a knowledge transfer tool, helping researchers and users improve tested prompt designs. Thus, this work not only systematizes prompt design but also provides a practical reference for prompt developers.

There are also common pitfalls in designing prompts based on AI behavior or output. Ronanki et al.[33] identify a set of problematic prompt patterns, referred to as prompt smells, which are recurrent issues in prompt formulation that can lead to undesirable or confusing outputs from generative AI models [2]. For instance, a convoluted or overly imaginative prompt—such as instructing the model to "imagine you are a Python snake calculating the area of a circle"—may result in correct output, but compromises explainability and interpretability. Similarly, vague prompts in a chain-of-thought setup, such as "write the area function" or "write the area parameters," lack coherence and make it difficult to trace which part of the prompt led to specific lines of generated code. Identifying and mitigating such prompt smells is crucial for improving the quality, relevance, and safety of LLM outputs, and contributes to more robust prompt engineering practices.

Although prompt patterns and catalogs are proposed, Ronanki et al. [34] introduce domain-specific prompt engineering guidelines. They contribute to a set of prompt engineering guidelines based on Requirements Engineering (RE) to improve the quality and reproducibility of LLM-generated outputs in software engineering contexts. Instead of focusing only on automatable techniques such as role-based prompting or step-by-step instructions, emphasis should be placed on improving the articulation of requirements through iterative feedback and refinement. Furthermore, they recommend decomposing complex requirements tasks into smaller, sequential prompts—a technique known as prompt chaining. This method helps to overcome input length limitations, maintain task focus, and ensure comprehensive coverage across activities such as requirements elicitation, classification, and specification.

To enable scalable and structured use of LLMs in practical applications, Cheng et al. [35] present a no-code framework that empowers users to build modular AI workflows using AI chains—sequential, prompt-driven components designed to perform complex tasks. Their system targets both developers and non-programmers by offering a visual interface for building and managing chains of prompt-based operations such as summarization, classification, and validation. Each step in the chain can be independently configured based on task-specific

requirements and expected outputs, supporting a high degree of modularity and flexibility. The framework also incorporates chat-based requirement analysis, enabling users to define goals in natural language and iteratively refine them into executable workflows. By embedding software engineering principles like modular design, reuse, and pattern catalogs, Cheng et al.'s approach facilitates the structured development, testing, and maintenance of scalable LLM-powered services. This aligns with broader prompt chaining strategies by promoting transparency, reusability, and collaboration in AI-driven application development.

2.5 LLM for Annotation

In previous approaches to metadata generation, manual data annotation has posed significant challenges particularly when annotators lack domain expertise. Accurately identifying elements such as model types, training strategies, or up-to-date methodological details across diverse academic domains requires both technical familiarity and contextual understanding, making the process labor-intensive and error-prone. Furthermore, manual verification of such metadata by human evaluators becomes increasingly infeasible at scale. To address these limitations, recent studies have explored using LLMs to automate data annotation, especially for linguistically and semantically complex tasks.

To explore LLMs in the context of data annotation, Tan et al.[36] provide a comprehensive survey that systematically analyzes how LLMs can be utilized for both annotation and synthesis tasks across diverse domains. The paper categorizes LLM-driven annotation into multiple forms such as instruction-response generation, rationale construction, pairwise and textual feedback and proposes quality criteria like diversity, informativeness, and consistency. It also reviews evaluation strategies for selecting high-quality outputs from multiple LLM generations and discusses how such annotations can be used for supervised fine-tuning, alignment training, and inference-time conditioning. The survey highlights key challenges such as hallucinations, annotation inconsistency, and domain transferability, while also addressing broader ethical concerns including bias propagation and transparency. This work provides a foundation for scaling annotation with LLMs and presents a guidance for incorporating synthetic annotations into modern ML pipelines.

Yadav et al.[37] examine the use of LLMs for automating complex linguistic annotation, using semantic proximity as a case study. They show how GPT-3.5 and GPT-4 can be guided through structured prompts based on annotation guidelines and partial examples to assess semantic relatedness between word usages across contexts. Their results underscore the sensitivity of LLM-based annotation to prompt design and show that even minor refinements or inclusion of annotated examples can substantially improve performance. The study also integrates these methods into an open-source annotation tool, demonstrating the feasibility of hybrid human–LLM annotation workflows. While promising, the findings emphasize that high-quality annotation still requires iterative prompt tuning and may benefit from human-in-the-loop mechanisms, especially for domain-specific or ambiguous tasks.

CHAPTER 3

METHOD

This work proposes a structured, question-driven pipeline designed to improve the explainability of paper-specific peer review generation. Rather than focusing solely on producing final reviews, this study emphasizes the process of capturing reviewer intent, aligning it with new submissions, and guiding review generation through semantically grounded prompts. There are two main stages in methodology: the preprocessing phase, and review generation phase as shown in Figure 1.

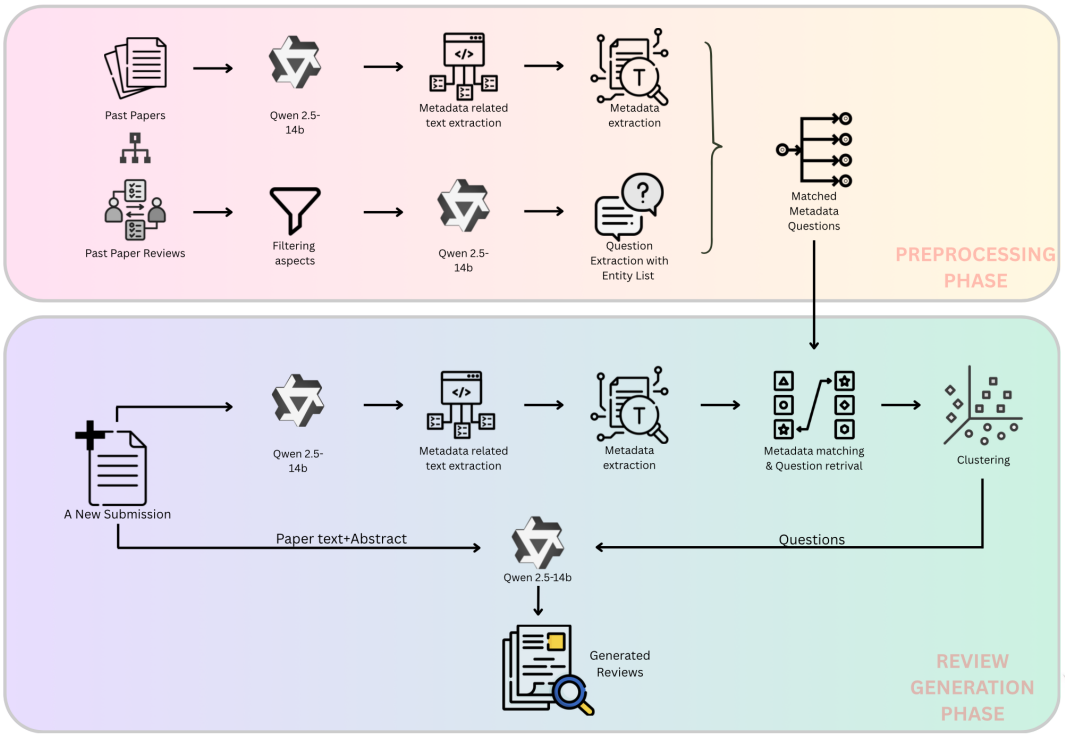


Figure 1: Two main phases of proposed methodology.

In the preprocessing phase, metadata extraction from past submissions is firstly performed. Then, entity list is generated for question generalization in metadata level with pilot study. Next, reviewer question extraction and generalization are conducted to build a reusable pool of domain-relevant reviewer questions. These questions are then aligned through question

matching with metadata embeddings, ensuring semantic relevance. To maintain focus, coverage filtering is applied, and a final set of representative questions is selected for each past paper.

In the review generation phase, the process begins with new submission metadata extraction, after which question retrieval and clustering are performed to align the past paper questions to the specific context of the submission. Finally, these curated questions are used to guide review generation, ensuring that the produced review remains contextually appropriate, evaluative, and domain-specific.

The practical guideline for applying the proposed methodology is provided in the Appendix A.1.

3.1 Dataset and Exploratory Data Analysis

In this section, I present the datasets used in this study and the exploratory analyses that informed subsequent methodological choices. The source and structure of the peer review corpus are first introduced, including the paper metadata, review texts, and aspect-level annotations. An analysis of the relationship between review aspects and acceptance decisions is then provided, highlighting the most influential dimensions for question generation and evaluation. These observations directly shaped the filtering and preprocessing strategies adopted in the proposed pipeline. Finally, the steps taken to clean, filter, and enrich the data resulting in a domain-specific, task-ready dataset, are described in detail.

3.1.1 Paper and Review Data

This study utilizes the ASAP-Review dataset, introduced by Yuan et al. [38], which compiles peer reviews and metadata from submissions to major machine learning conferences. Specifically, the dataset covers papers and their reviews from ICLR from 2017 to 2020, and from Conference on Neural Information Processing Systems(NeurIPS) from 2016 to 2019. Since NeurIPS only publishes reviews for accepted papers, I focus on the ICLR subset, offering a more balanced dataset in terms of accept/reject ratio, which is important for analyzing review criteria objectively.

Each entry in the dataset includes the following components:

- ***Paper Metadata:*** Title, authors, abstract, and full body parsed using the AllenAI Science-Parse tool.[39]
- ***Review Texts:*** Reviews from program committee members, meta-reviews from area chairs, and the final decision label (accept or reject), confidence score of reviewer, rating to paper.

The summary statistics for the dataset is provided in Table 1.

The ASAP-Review dataset is used in various tasks, primarily from an aspect–peer review perspective. In subsequent years, the authors of the dataset, Yuan and Liu [40] studied

Table 1: Basic statistics of ASAP-Review dataset for ICLR proceedings.

	Accept	Reject	Total	Avg. Full Text Len.	Avg. Review Len.	# of Reviews
ICLR	1,859	3,333	5,192	7,398	445	15,728

also on review generation and introduced the Kid-Review model, which improves external knowledge to produce more comprehensive reviews. The method fine-tunes a BART model by incorporating citation embeddings and a concept graph encoded via a Graph Attention Network (GAT). This process is framed as an aspect-based summarization task, aiming to generate reviews that both recommend a paper's publication status and provide targeted feedback. Rimkus et al.[41] present a comparative study of machine learning models to predict paper acceptance at the ICLR conference. Using the same dataset of ICLR papers from 2017 to 2020, the authors compare traditional machine learning models trained on handcrafted features with transformer-based models. Handcrafted features include the presence of specific keywords in the abstract, paper metadata such as title length and number of authors, and structural elements like the number of figures and sections. The study found that a Multi-Layer Perceptron (MLP) model, using these handcrafted features, achieved a slightly higher accuracy (78.37%) than the best-performing transformer model, Longformer (76.92%). Kumar et al.[42]also proposed a model including three distinct channels of information: the raw peer review text, the specific aspect category being discussed (e.g., novelty, soundness), and the sentiment associated with that aspect. Using the ASAP-Review dataset of ICLR and NeurIPS papers, the authors developed a generic and adaptable model. Their findings show that incorporating aspect information alongside sentiment improves accuracy by approximately 3.3 points, with their model achieving a competitive accuracy of up to 76.67%. This work highlights the value of using sentiment and aspect information to create a controllable system for better editorial decision-making, while also acknowledging limitations like contradictory reviews and the potential for a larger dataset. Muangkammuen et al.[43] also studied on ASAP-Review dataset to address the challenge of predicting specific peer-review aspect scores (e.g., "clarity," "originality") despite limited public datasets. The authors propose an intermediate-task transfer learning method, which involves fine-tuning a pre-trained model (SCIBERT) on a related intermediate task before training it on the final score prediction task. Using the ASAP-Review dataset, the authors found that this transfer learning approach improved prediction performance across all target tasks, with an average increase of 4.1% in accuracy and 8.4% in macro F1 comparing to baseline. The most significant improvements were observed for the "Soundness/Correctness" score prediction. The study concludes that this method is effective in improving the performance of peer-review score prediction models.

3.1.2 Aspect-Based Annotated Data

The ASAP-Review dataset models the multifaceted nature of peer reviews by including aspect-level annotations adapted from ACL review guidelines. Each review sentence is labeled with one or more of the following aspects: *Summary (SUM)*, *Motivation/Impact (MOT)*, *Originality (ORI)*, *Soundness/Correctness (SOU)*, *Substance (SUB)*, *Replicability (REP)*, *Meaningful Comparison (CMP)*, and *Clarity (CLA)*.

Sentiment polarity annotations (positive or negative) are included for all aspects except *Summary* in the original dataset. This study does not utilize polarity information since sentiment analysis falls outside the scope of this work.

The aspect annotations originate from the original dataset authors [38], who applied a rigorous pipeline involving manual annotation, BERT-based model training, and heuristic post-processing. These annotations enable structured analysis of peer reviews and provide the foundation for question extraction in this study. The work relies solely on the aspect labels, without performing additional re-annotation or sentiment classification.

3.1.3 Statistical Analysis of Aspect–Acceptance Relationship

To prioritize the types of questions to be generated, I conducted an analysis of aspect distributions and their relationship with review outcomes. The goal of this analysis was to identify which aspects from the ACL review checklist are most predictive of paper acceptance, so that these aspects could be emphasized during question selection and prompt construction. The dataset used for this analysis includes reviews from ICLR (2017–2020) dataset [38], each tagged with one or more aspects from the ACL checklist: Summary, Motivation/Impact, Originality, Soundness/Correctness, Substance, Replicability, Meaningful Comparison, and Clarity.

To investigate the relationship between review aspects and paper acceptance, three complementary statistical analyses were performed:

1. **Logistic regression** was used to estimate the effect size (coefficient) of each aspect on the probability of acceptance, along with its statistical significance.
2. **Two-sample t-tests** compared the mean reviewer ratings for accepted vs. rejected papers, testing whether the presence of each aspect is associated with significant rating differences.
3. **Chi-square tests of independence** assessed whether the presence of each aspect is independent from the final acceptance decision.

The null hypothesis in all cases stated that the aspect has no effect on acceptance probability or reviewer ratings, and is independent from acceptance outcome. Tests were conducted at the 5%($p < 0.05$) significance level.

Logistic regression results indicate that *motivation* has the largest positive coefficient (0.0108, $p < 3.24 \times 10^{-24}$), implying that its presence increases the likelihood of acceptance. *Clarity* and *originality* also show positive effects, though *originality* is not found as significant in the t-test, suggesting a weaker link to reviewer ratings. Conversely, *comparison* has the most negative coefficient (-0.0125, $p < 2.02 \times 10^{-41}$), indicating that the absence of this aspect is strongly associated with rejection. Negative but significant effects are also observed for *substance*, *replicability*, and *soundness*. Chi-square and t-test results largely align with the regression findings, reinforcing the conclusion that most aspects, except for *originality*,are consistently associated with acceptance outcomes and reviewer scores as shown in Table 2.

Table 2: Combined statistical results for each aspect, sorted by logistic regression coefficient. Significant results ($p < 0.05$) are in bold.

Aspect	Coefficient	LogReg p	t-test p	χ^2 p	Significant?
motivation	0.010796	**3.24e-24**	**4.49e-22**	**6.63e-10**	Yes
clarity	0.002953	**4.10e-05**	**5.11e-04**	**8.73e-11**	Yes
originality	0.002573	**2.76e-03**	6.52e-02	**1.92e-05**	Partial
summary	0.001421	**5.67e-13**	**5.54e-13**	**1.67e-16**	Yes
soundness	-0.003918	**1.37e-08**	**2.28e-10**	**1.20e-04**	Yes
substance	-0.007111	**7.47e-17**	**2.63e-23**	**1.58e-09**	Yes
replicability	-0.009988	**5.02e-09**	**8.74e-11**	**2.17e-05**	Yes
comparison	-0.012502	**2.02e-41**	**2.11e-50**	**3.04e-32**	Yes

To assess each aspect's influence on accept/reject decisions, a logistic regression model was trained with Accept encoded as 1 and Reject as 0. The data was split into 80% training and 20% testing sets. The model learned coefficients for each aspect: positive values indicate higher acceptance likelihood, negative values indicate higher rejection likelihood. While logistic regression shows the direction of influence, a Random Forest classifier captures complex, non-linear relationships through feature importance scores. A Random Forest classifier is trained on the review data with `n_estimators=100` and `random_state=42` for reproducibility. Feature importance scores are extracted to measure each aspect's impact on the accept/reject prediction. The aspects are ranked by importance and visualized in a bar chart. Scores are always positive, with higher values indicating greater influence on the accept/reject decision. Sorting these scores ranks aspects from most to least influential (See Fig. 2).

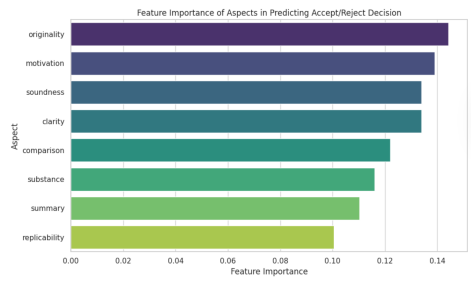
- A positive coefficient indicates that an aspect makes a paper more likely to be accepted.
- A negative coefficient suggests that an aspect makes a paper more likely to be rejected.

Paper reviews from 2017 to 2020 are also examined to determine the presence or absence of specific aspects. Figure 3 shows which aspects were most frequently mentioned in reviews over the years and which ones demonstrated an increasing trend. This distribution is further illustrated in a histogram showing the frequency of different aspect coverage levels, where the x-axis represents the aspect coverage and the y-axis represents the frequency of reviews at that coverage level. The calculation of aspect percentages is performed using the following formula:

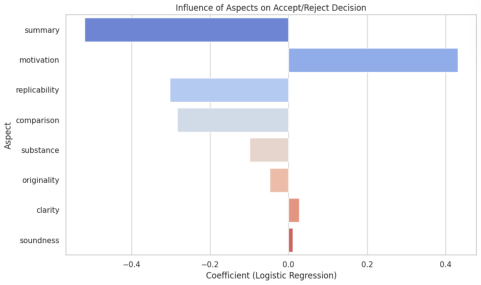
$$\text{Aspect Percentage} = \frac{\sum_{i=1}^{N} \mathbb{I}(\text{Aspect}_i)}{N} \times 100$$

where:

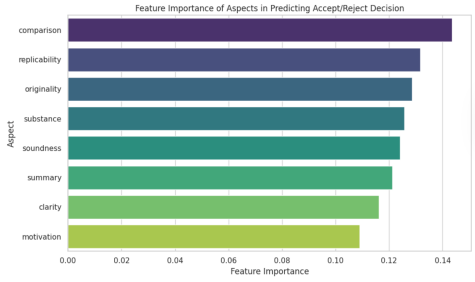
- N is the total number of paper reviews in each year
- $\mathbb{I}(\text{Aspect}_i)$ is an indicator function that equals 1 if the aspect is present in review i and 0 otherwise. This is represented by the sum of a specific aspect column, and counts the total number of reviews where that aspect was mentioned.



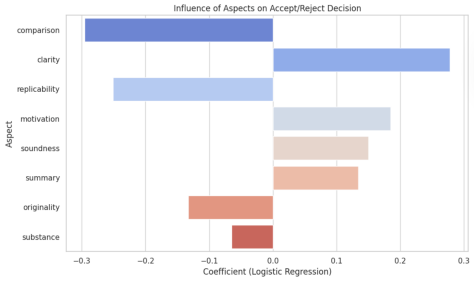
(a) Feature importance (2017)



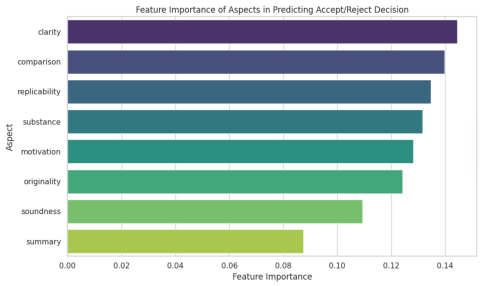
(b) Aspect coefficients (2017)



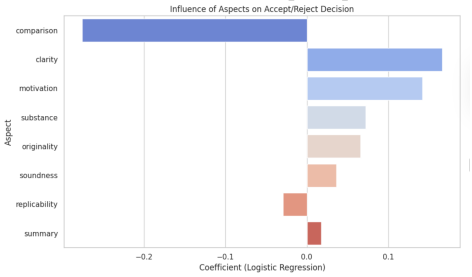
(c) Feature importance (2018)



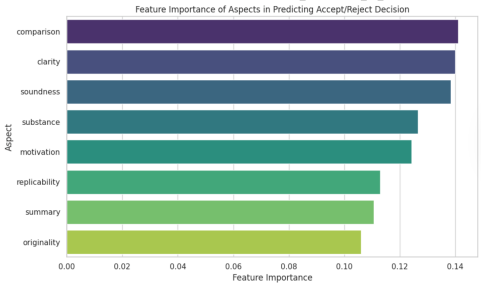
(d) Aspect coefficients (2018)



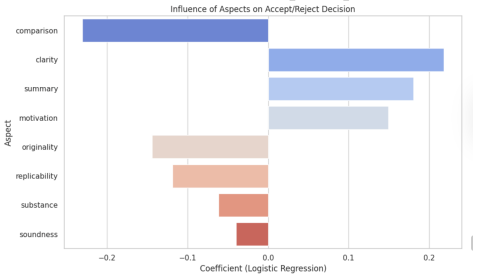
(e) Feature importance (2019)



(f) Aspect coefficients (2019)



(g) Feature importance (2020)



(h) Aspect coefficients (2020)

Figure 2: Feature importance and aspect coefficients for acceptance/rejection decisions by year.

This formula computes the percentage of reviews that mention a particular aspect out of the total number of reviews.

In terms of coverage of each aspect, Clarity and Soundness were the two most dominant aspects, appearing in more than half of all paper reviews each year (See Fig. 3). The least frequent aspects, Comparison and Replicability, remained constant in terms of their raw mention counts, but their proportion of total reviews increased over the period. Overall, the distribution of aspect coverage in the reviews showed an average coverage percentage of approximately 0.50.

The results in Figure 2 demonstrates the importance of Comparison and Clarity aspects on accept-reject decisions for all the years except 2017 submissions while Figure 3 shows that replicability is the least covered aspect in overall reviews for all four years although it meant to be covered by reviewers. Based on these findings, the aspect-level insights were incorporated into multiple stages of the proposed system pipeline as follows:

- **Question Generation:** Only reviews labeled with *clarity*, *comparison*, and *replicability* aspects were selected and preprocessed to generate training-time question-answer pairs. This ensured that the system focused on generating questions grounded in impactful and informative review content.

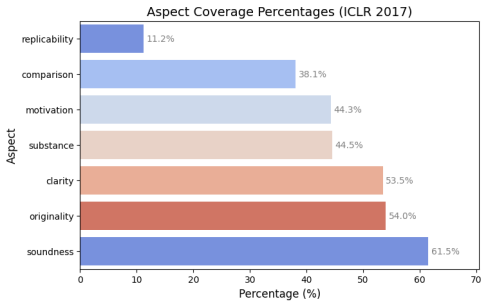
- **Test Filtering and Validation:** For evaluation on test set(ICLR 2020 Submissions), test reviews were filtered to extract only sentences tagged with the same three aspects. Questions were then generated from these selected spans, and validation was performed by matching them against pre-generated question clusters derived from the training data.

- **Generated Review Evaluation:** During the comparison of generated reviews with golden human-written reviews, only sentences corresponding to the selected aspects were retained from both sides. This allowed the evaluation to focus not on generic or high-similarity phrases, but on fine-grained, aspect-specific content that could be directly and meaningfully compared.

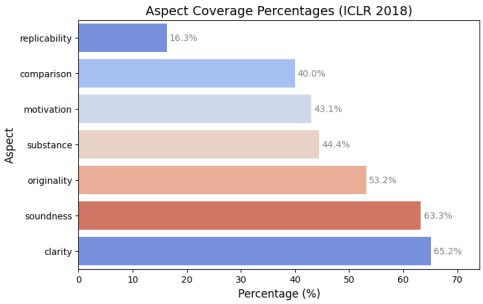
Through this selective and aspect-aware strategy, the system was aligned to generate, match, and evaluate content specifically around the most influential dimensions of peer review, rather than relying on overall textual overlap.

These observations are supported by prior literature. Shruti et al.[44] reported that clarity was a strong predictor of acceptance, and that replicability, though underrepresented, played an important role in high-quality reviews. Reviews that compared methods or baselines were more detailed and often linked to accepted papers.

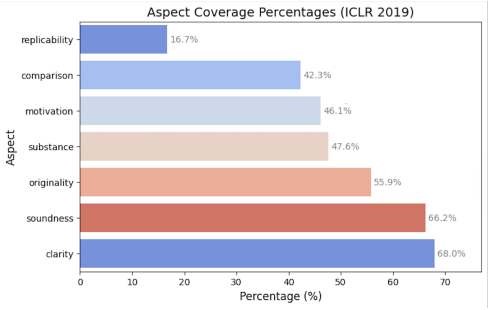
Findings from Yuan et al. [38] also supported this result. Their work highlighted clarity as the most predictive aspect for acceptance. Replicability was also found to be important, despite being less common.



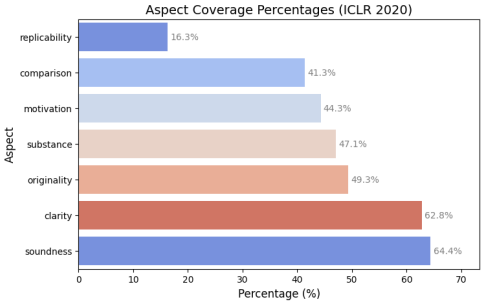
(a) 2017



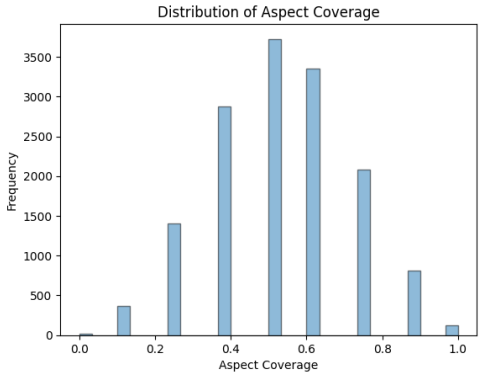
(b) 2018



(c) 2019



(d) 2020



(e) Total Coverage

Figure 3: Aspect coverage distribution for each year and overall total.

3.1.4 Preprocessing

This section describes the preprocessing steps applied to construct a task-specific dataset suitable for question extraction and review generation experiments.

In the original dataset, it lacks complete metadata (e.g., abstracts, paper titles), additional information was obtained from the OpenReview platform. To complete missing metadata, abstracts were retrieved through the OpenReview API. Paper IDs were matched with official ICLR records from 2017 to 2020, and missing fields such as titles and abstracts were appended to the dataset. This ensured metadata consistency across all entries used in the experiments.

The original paper texts were segmented by section headers, and only sections relevant to review criteria—such as methodology, experiments, and results—were retained. Sections like *Related Work, Background, and References* were excluded to focus on content that contributes directly to evaluating clarity, comparison, and replicability.

Reviews were linked to their corresponding papers through a consistent identifier scheme, enabling reliable pairing for analysis. This restructuring ensured consistent alignment between reviews and their associated papers, facilitating their use as paired inputs in question extraction, matching, and review generation steps. The ASAP-Review dataset already contained basic paper-level metadata such as title, authors, abstract, parsed full text, reviews, decisions, reviewer confidence, and ratings. While this information was useful for understanding the dataset's scope, it primarily served as general descriptive content and did not offer content-specific attributes needed for metadata-level match of papers. For the main experiments, I instead relied on task-specific metadata generated via LLM-based extraction from section-specific paper text. This second metadata category was explicitly designed to capture features relevant to both the paper's technical content and its potential alignment with reviewer questions. The extracted features were: **Model** — the main algorithmic or architectural approach proposed or used (e.g., transformer-based encoder, graph neural network). **Dataset** — the dataset(s) employed for training and/or evaluation (e.g., CIFAR-10, SQuAD, ImageNet). **Evaluation metric** — the performance measures used (e.g., accuracy, BLEU, F1 score, perplexity). **Training strategy** — learning setup, including techniques such as curriculum learning, supervised learning, unsupervised learning, etc. **Data type** — the modality of the input (e.g., text, image, audio, multimodal).

3.2 Prompt Construction

The prompt construction process was informed by an initial pilot study, which evaluated the feasibility of extracting paper metadata and review questions without using an LLM. Insights from this phase guided the design and refinement of the subsequent prompting strategies. At this stage, metadata were manually extracted from a selected set of papers, and sample questions were generated by analyzing the corresponding reviews to identify recurring patterns and reviewer concerns. The resulting metadata and questions served as a baseline for validating future outputs on a stratified subset of the dataset.

A similar multi-step prompting strategy was adopted for the extraction of paper-specific features. Rather than using a single prompt to extract all the metadata features for paper alignment (model, dataset, evaluation metric, training strategy, and data type), the process was divided into separate prompt chains, with each one dedicated to a single feature. This decomposition improved the focus and precision of the model's outputs and minimized overlap or confusion between related features.

3.2.1 Pilot Study

The initial phase of prompt construction focused on a pilot study and selection of papers from ICLR 2017 proceedings. This was needed to properly structure LLM prompts. Preference was given to submissions employing applied methodologies with implemented deep learning models, rather than purely theoretical contributions. A total of 45 papers (See Appendix A.5) were selected using a stratified random sampling approach, consisting of 30 accepted and 15 rejected papers, to ensure balanced representation across decision outcomes. A manual review of 45 papers was conducted to distinguish theoretical work—characterized by proofs or mathematical derivations without empirical results—from applied studies. This distinction was based on the presence of: (i) equation- or proof-based experiments with numerical results, or (ii) implemented deep learning models evaluated with standard or proposed metrics. In addition, each paper was manually inspected to verify if it used text/image/other data. To extract dataset and data type information, preprocessed paper texts were provided to Qwen2.5-14b. This model was selected because it can be run on local machines, does not produce explicit reasoning steps in its outputs, unlike LLMs such as DeepSeek R1, and is computationally affordable. The output was expected to return only the metadata content rather than reasoning. The model was expected to infer which dataset was used in experiment for training or for evaluation and predict the data type accordingly. Among the 45 papers, the model correctly predicted the data type for 36 papers, yielding 80% accuracy. In 5 of the 9 incorrect predictions, errors were due to the extraction of incorrect dataset names. The remaining 4 errors were caused either by failure to extract any dataset or misclassification of the data type despite correct extraction.

In addition to metadata extraction, I manually examined the reviews of papers (45 papers and 138 reviews), from which I identified 442 questions that are considered by reviewers while giving comments for the papers. These served as the foundation for the LLM prompts described in the following sections. The extracted questions reflected how a human annotator would identify review questions or metadata from paper text, and how such questions could be generalized. I used this annotated set to test the LLM's ability to replicate human reasoning—acting as a "human attention mechanism"—by providing prompts enriched with relevant details and nuanced examples derived from this study. Selected prompt trials are the ones utilized during experiments on ICLR 2019&2020 submissions for both metadata extraction, review questions generation and post processing steps.

3.2.2 Prompt Construction For Metadata Generation

The metadata generation prompts were tested in an ad-hoc way with varying input scopes and formats to improve extraction accuracy as demonstrated in Table 3. Zero-shot prompts on full texts or abstracts often produced incomplete outputs. Data types list consists of text,

image, and other types of data. Full paper text excluded related/background work, references and appendix sections. Taxonomy list was the 2025 taxonomy list offered by IEEE as July 2025 IEEE Taxonomy for Computational and Artificial Intelligence[45] (See Appendix A.3). The best results were the selected ones while comparing with manually extracted metadata features, and the best performing LLM results were obtained using a template-based prompt with filtered, paper-relevant paragraphs including paper, enabling consistent and structured JSON outputs. All the final prompts for paper content relevant text and metadata extraction are available in Appendix A.2.5.

Table 3: Prompt Trials for Metadata Extraction

ID	Format	Content	Output Format	Selected
P1	Zero-shot	Full paper text, abstract, taxonomy list, data types list, prompt	JSON	✗
P2	Zero-shot	Filtered paper text, abstract, data types list, prompt	JSON	✗
P3	Template + Zero-shot	Metadata-relevant text from paper, data types list, prompt	JSON	✓

3.2.3 Prompt-Response Workflow for Question Extraction, Generalization, and Post Processing

Key elements considered during question formation were first identified and listed. These elements included whether the reviewer referred to the method, dataset, comparison to prior work, or clarity of explanation. When each aspect was tested individually, the generated questions were generally relevant.

Combining all aspects into a single prompt and increasing the number of input papers resulted in noisy and less meaningful model outputs. To address this issue, a chained prompt structure was developed. In this structure, each step (or "chain") targeted a specific subtask within the broader goal of generating reviewer-style questions or extracting important paper features.

The chained approach contributed to greater consistency in responses and reduced ambiguity, particularly when processing lengthy texts or papers with heterogeneous structural characteristics.

Selected prompt trials for each task are utilized during experiments

Table 4: Prompt Trials for Reviewer Question Extraction Steps

ID	Format	Content	Task	Selected
P4	Zero-shot	Full raw review, abstract, long prompt	Question Extraction	✗
P5	Template-based	Abstract, filtered review, minimal prompt	Question Extraction	✓
P6	Zero-shot	Full raw review, abstract, minimal prompt	Question Extraction	✗
P7	System message	Full raw review, minimal prompt	Question Extraction	✗
P8	Zero-shot	Full raw review, long prompt	Question Extraction	✗
P9	Few-shot	Input questions, entity list, example, long prompt	Question Generalization	✓
P10	Few-shot	Input questions, entity list, example, minimal prompt	Question Generalization	✗
P11	System message + Few shot	Input questions, entity list, example, short prompt	Question Generalization	✗
P12	Zero-shot	Entity name replaced review, minimal prompt	Question Generalization	✗
P13	Zero-shot + Instruction-based	Questions, entity list	Question Post Processing	✓

There are multiple attempts to extract review questions from review text as shown in Table 4. The "Format" column indicates the prompting approach used: Zero-shot relies solely on task instructions without examples, Template-based uses predefined input structures to guide the model, Few-shot includes representative examples to illustrate desired outputs, and System message configures the model's behavior through role-based instructions. Combined formats (e.g., "System message + Few-shot") use multiple techniques to balance guidance and flexibility.

Each prompt trial was revised according to the observed errors or when model responses diverged from the desired target. An initial version included a long-form prompt containing all instructions. This prompt aimed to guide the model by providing examples of both acceptable and undesirable word groups. Reviews from the same paper were given as input alongside the prompt.

An example illustrating the prompt structure, input review, intended response, and model output is provided in Appendix A.2.1 and the final prompts for question extraction, metadata-level generalization, and post processing are available in Appendix A.2.2.

3.3 Metadata-level Generalization of Review Questions

Review questions in scientific peer review often contain highly domain-specific terms, limiting their reuse across diverse research papers. To address this, a method is proposed to systematically generalize these questions at the paper relevant text level by replacing specific domain terms with standardized placeholders called entity labels. These labels represent groups of semantically similar entities, the original domain-specific terms, forming an entity list. This approach enables creation of reusable, context-independent question templates that preserve their core meaning while improving scalability across topics.

3.3.1 Entity List Creation

This phase of the study focused on generalizing questions extracted from reviews using the entity name list. In this context, an entity or entity name refers to a paper-specific element such as a dataset, model, evaluation metric, training strategy, or domain-specific terminology. These entities represent the concrete details of a paper that can later be abstracted into higher-level placeholders. Generalization of questions refers to transforming reviewer questions so they are applicable beyond the specific paper, replacing paper-specific entities with metadata-level placeholders while retaining the question's evaluative intent. The pipeline was reversed: instead of extracting questions directly from raw reviews, reviews were first generalized by applying a predefined entity list, and questions were then extracted from these generalized, preprocessed reviews. The rationale behind this approach was that generalization could standardize and abstract key content, improving the quality of question generation. However, providing both the abstract and the full review to the model during generalization led to confusion and poor performance—key entities were often lost, and unrelated outputs were generated. Improved results were obtained when only a single sentence was processed at a time for generalization. To further improve this, multiple chunking strategies were explored, including sentence-wise splitting, fixed-length character segmentation, and overlapping chunks that preserved context by retaining part of the previous chunk. It was observed that none of the chunking methods significantly outperformed the simpler approach of directly inputting the raw review and abstract into the model for question extraction, followed by a separate generalization step applied to the extracted questions. The method was applied to review questions extracted by Qwen2.5-14b from 45 papers and their 138 reviews from the pilot study, identifying 381 entity names in brackets. After 12 iterations, a final entity list of 54 representative entity names was produced (See Appendix A.3).

Qwen2.5-14b model was instructed to replace domain-specific terms in questions with bracketed placeholders. The following example illustrates this process:

- Original: "Q1: Why is recursion considered important for neural programming architectures?"

- Generalized: “GQ1: Why is [main concept] considered important for [domain]?”

The entity list creation process followed the structured workflow shown in Figure 4.

The generalized extracted review questions were manually inspected. The generalized outputs included a large number of vague or overly broad expressions. To enable systematic generation of reusable and context-independent review questions, the construction of an entity list was required.

Most review questions contained domain-dependent terms, which limited their reusability across different submissions. While such terms were specific to the domain of the dataset in this study, the objective was to abstract them at the metadata level to enable broader applicability. Without this abstraction, question templates would become too detailed to match equivalent questions in other domains, or failing to align with unseen data when a new model or dataset is introduced. Terms with similar meanings were grouped under representative entity labels, and group validity was confirmed through cross-question checks to ensure semantic coverage. Multiple annotators resolved cases where label assignment was uncertain, including repeated terms, conflicting meanings, or terms whose interpretation could vary depending on context. To illustrate, for the question:

- Why is recursion considered important for neural programming architectures?

terms like “recursion” and “neural programming architectures” were replaced with entity placeholders [main concept] and [domain]. If semantically similar terms already existed in the entity list (e.g., “related work” or “field”), group validity was used to decide inclusion. Decisions such as whether [main concept] could be replaced with [proposed method] were discussed for semantic relevance. The final entity list is given in Appendix A.4 was reviewed by an NLP expert working as teaching assistant in the NLP domain in METU and independently checked by two other annotators: a software engineer working in deep learning and a computer engineering master’s students, both of whom METU graduates and familiar with deep learning applications, and by myself as an additional annotator.

3.3.2 Iterative Generalization and Marking

The construction of the entity list followed an iterative cycle involving question generation, identification of domain-specific expressions, semantic grouping, and systematic refinement. Initial review questions were derived from selected papers, each containing terminology strongly linked to the respective research context. Generalization was achieved by marking domain-dependent terms using square brackets, however; they are usually free-style and random. Thus, a semantic analysis was performed by annotators with group validity after the marking phase to identify synonymous or semantically similar expressions across multiple questions. These expressions were grouped under unified entity labels, laying the foundation of a structured entity list. For instance, terms like [approach], [main concept], or [core methodology] was decided to be represented by entity name [concept] by group validity.

The iterative nature of the process is important: each iteration revealed new domain terms or inconsistencies in current groupings, which are then addressed. This cycle continued until the

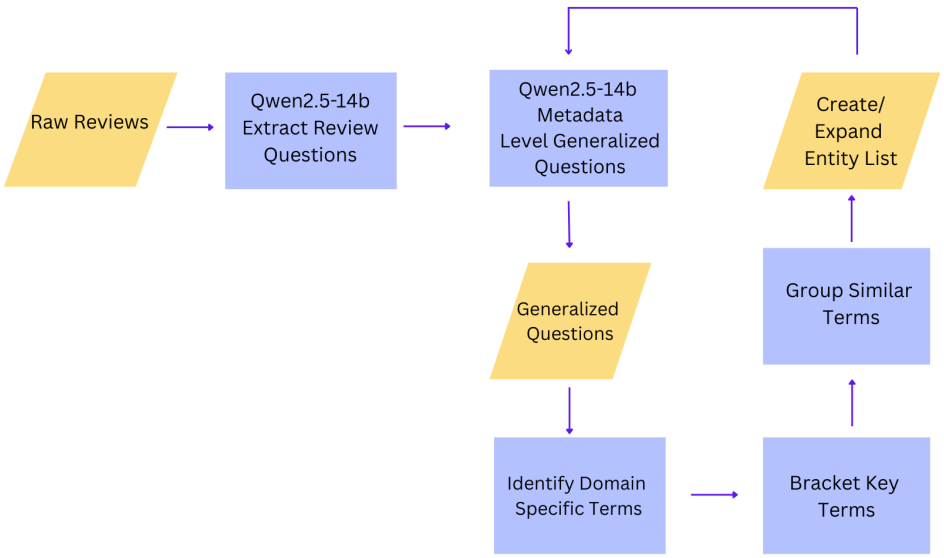


Figure 4: Workflow for LLM-assisted Generation of a Structured Questions and Entity List.

entity list reached a stable, comprehensive state. Entity definitions were revised based on the requirement that generalized questions must remain informative yet sufficiently flexible to be reused across various paper contexts without manual intervention.

A concrete example from the entity construction procedure is provided in the following to illustrate the applied methodology:

- *"Q1: What specific evaluation metrics were used to compare the variational autoencoder-based generative model with a standard autoregressive decoder?"*

The phrase "variational autoencoder-based generative model" was identified as a domain-specific term. Initially, this expression was mapped to the general entity label ***model*** and included in the entity list.

Subsequent analysis revealed an additional domain-specific phrase within the same question: "autoregressive decoder." Unlike the previously identified model, this term referred to a baseline system rather than the proposed approach introduced in the paper.

To reflect this contextual distinction, separate mappings were introduced:

- "variational autoencoder-based generative model" → ***proposed model***,
- "autoregressive decoder" → ***baseline/compared model***

Both entities were independently added to the entity list to ensure semantic clarity between the compared systems.

This case underscores the importance of semantic context in the generalization process. Iterative refinement of entity labels was necessary to preserve meaningful distinctions across structurally similar, yet contextually different terms.

Terms with similar meanings (e.g., "evaluation metric" and "performance measure") were merged into a single label to maintain applicability despite terminology variations. After each generalization cycle, problematic terms were revised manually to avoid random entity name generation of the LLM. This recursive refinement produced a comprehensive, adaptable entity list for diverse domain-specific review questions.

3.4 Matching Reviewer Questions via Metadata Embeddings

To evaluate the method in a realistic and fair setting, I focused on an unseen test dataset. The goal was to assess how well questions could be generated and selected for a new paper based solely on its metadata, without access to its full content. This approach implies domain-relevant review questions for a paper are inferred from previous papers with similar characteristics.

Reviewer questions from previously reviewed papers were retrieved using a similarity-based matching approach applied to dense vector representations of paper metadata. In this study, dense vector representations refer to embedding-based encodings of text, where semantically similar inputs are placed closer together in a high-dimensional space [46]. The similarity between metadata fields was used as a proxy for the likelihood that a paper would receive similar reviewer questions. Each paper was described using five metadata fields: Training Strategy, Data Type, Dataset, Evaluation Metric, and Model Type. These fields were concatenated into a single textual string.

Embedding and Indexing. All metadata strings in the reference set were encoded with the Sentence-BERT model `all-MiniLM-L6-v2`. Separate FAISS(Facebook AI Similarity Search)[47]indices were created for each pairwise combination of metadata fields (total 10 combinations). FAISS is a library for efficient similarity search and clustering of dense vectors at scale. Higher-order combinations were not explored due to sparsity. These indices enable efficient nearest-neighbor search in embedding space to identify semantically similar papers (See Fig. 5).

Similarity-Based Matching. For each test paper, metadata strings matching each field combination were encoded using the same model. The top-10 nearest neighbors were retrieved via the corresponding FAISS index, and their reviewer questions extracted. This procedure was repeated for all 10 metadata field pairs, storing matched questions per paper and per combination.

Coverage Aggregation Across Combinations. Each paper from the test set can match various retrieved relevant questions based on different metadata combinations with past papers. Therefore, coverage was defined as the proportion of matched questions exceeding a similarity

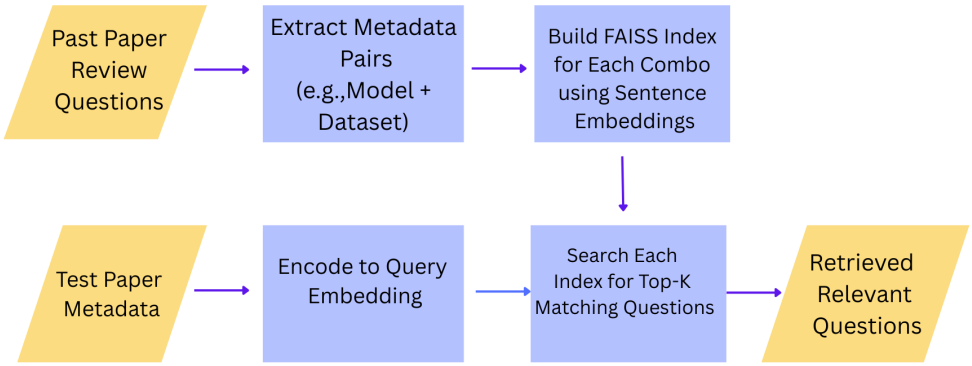


Figure 5: Matching unseen papers with prior questions using FAISS indexes built on metadata combinations

threshold of 0.5 in ad-hoc manner. These scores were averaged across combinations. This approach allows integration of information from multiple paper-relevant text perspectives (See Fig. 6).

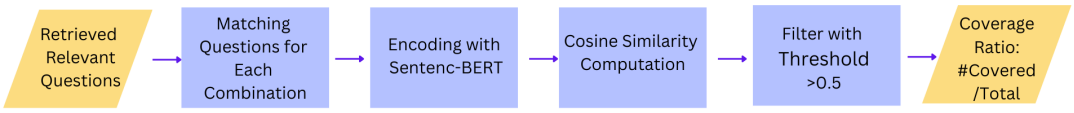


Figure 6: Computation pipeline for semantic coverage ratio using matched questions.

Representative Question Selection. A diverse and informative subset was selected from all matched questions across metadata combinations with a similarity score above 0.5. However, these questions could still be semantically similar to each other. For example, [How is the proposed method compared with baseline method?] and [How do you compare the proposed framework with another work?] get almost the same answer from the paper text. Therefore, even with high similarity scores, the goal here was to form clusters that are as semantically different as possible. For this reason, a diverse and informative subset was selected from all matched questions across metadata combinations. Question embeddings were clustered using K-means clustering, and the optimal number of clusters ($K = 12$) was determined by the elbow method. From each cluster, the question closest to the centroid was chosen as the representative (See Fig. 7).

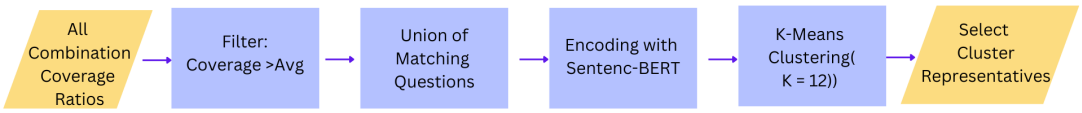


Figure 7: Representative question selection pipeline using clustering over high-coverage question combinations.

CHAPTER 4

EXPERIMENTS

This chapter presents the experimental setup, quantitative, and qualitative analysis results. The experimental setup section presents computational resources, baseline model selection, and LLM-based review generation with its evaluation process. The results cover the effectiveness of metadata-guided question matching, comparisons with generated reference questions, and review generation versus gold reviews using various models and evaluation criteria. Sentence-level evaluation is also conducted to address the limitations of review-level metrics. The qualitative study details question set diversification.

4.1 Experimental Setup

4.1.1 Computational Resources

The experiments were carried out on two distinct computing environments. The primary server was equipped with four NVIDIA A100 80GB PCIe GPUs, offering substantial memory and computational capacity, which enabled the processing of all papers in dataset in the main evaluation experiments. For smaller-scale subsets, prompt design, and POC study, a secondary server with an NVIDIA GeForce RTX 4070 Ti GPU (32GB) was employed. This configuration allowed efficient allocation of resources according to the computational demands of different stages in the pipeline, balancing throughput with accessibility.

4.1.2 Initial Selection of Baseline model

This section evaluates the models to find baseline models, in which each baseline-candidate LLM is prompted to generate a review solely based on the provided paper content, without any metadata guidance or targeted reviewer questions. The study benchmarks the metadata-question-based review generation pipeline by evaluating reviews produced with five LLMs alongside the model used in the experiments: Qwen2.5-14B, Qwen3-14B, gemma3:12b, llama3:70b, and deepseek-r1:latest, each following the same input setup. As context, the full paper text including the abstract and main body was supplied. The exact prompt used for all baseline runs was:

"You are a conference reviewer for the top tier ICLR. Consider the paper in terms of clarity, comparison and replicability, then evaluate it. Given the paper's abstract, full content, write

a critical and objective review for the paper. Your tone should be professional, neutral, and objective, not overly positive or promotional. Be to the point. Avoid meta-statements. Output ONLY the final review text.
Abstract:
{abstract}
Paper Content:
{text}
Generated review: "

All experiments for baseline selection were conducted on the set of 100 unseen papers among ICLR 2020 test set to manage computational costs. These papers were selected with stratified sampling method based on acceptance (50 accepted - 50 rejected) and do not include pilot study papers. Limiting the evaluation to 100 papers enabled a feasible yet meaningful comparison of model performance for baselines. The best BERTScore [48] and embedding similarities [46] were the metrics used for baseline model selection. Qwen3-14B(qwen3_b1_v1) demonstrated the best trade-off between semantic similarity and execution time efficiency, making it the most suitable choice for further experiments, as shown in Table 5, and was therefore selected for further use. The Qwen2.5-14b(qwen2.5_b2_v1) is also selected as baseline since it outperforms other models in embedding similarity score. While DeepSeek-R1 also produced competitive results, it was not selected as a baseline model. Similar to the pilot study design, the preference was given to models that provide comparable performance while being computationally less expensive, operable under limited local resources, and configurable to disable reasoning steps. This ensured consistent outputs without requiring additional post-processing, making Qwen3-14B and Qwen2.5-14B more practical choices for downstream experiments.

Table 5: BERTScore, embedding similarity, and execution time for models on review generation without reviewer questions on 100 papers

Model	BERTScore	Embedding Similarity	Execution Time
qwen2.5_b2_v1	0.580	0.683	21m45s
qwen3_b1_v1	0.598	0.650	28m56s
Gemma3-12B	0.560	0.648	13m36s
LLaMA3-70B	0.324	0.046	53m22s
DeepSeek-R1	0.595	0.641	14m56s

4.2 Results and Evaluation

The evaluation covers two complementary strategies. Firstly, for unseen test papers, metadata extracted from the paper content is used to automatically retrieve applicable reviewer questions via the proposed method. These questions are compared with relevant ones extracted from the actual reviews to measure overlap. Secondly, without using the test set reviews directly, retrieved questions are matched with the paper content to generate review sentences, which are then compared to actual reviewer sentences at fine granularity (e.g., sentence level, review level), including baseline comparisons. Evaluation aims to quantify how well the method

retrieves and applies relevant questions and how closely the generated reviews align with human-written ones. The evaluation results were achieved by implementation of both strategies on all the papers in ICLR 2020 test set among ASAP-Review dataset.

4.2.1 Effectiveness of the Proposed Metadata-Guided Question Matching

The proposed metadata-guided matching as explained in Section 3.4 was evaluated based on its ability to cover review questions extracted from ground truth reviews by the proposed method. Semantic coverage was computed using cosine similarity over Sentence-BERT embeddings, with a threshold of 0.5 applied in an ad hoc manner to determine whether a retrieved question matched a generated question. Reviewer questions with semantic similarity above the threshold were classified as covered. Finally, the mean semantic coverage of retrieved questions for each pairwise combination was calculated and is reported in Table 6.

Table 6: Mean coverage values of the best covering pairwise metadata feature combinations.

Best Covering Pairwise Combination	Mean Coverage
dataset+data type	0.669
dataset+model	0.671
dataset+training	0.622
dataset+evaluation metric	0.665
model+evaluation metric	0.641
training+evaluation metric	0.615
data type+evaluation metric	0.697
data type+training	0.592
data type+model	0.715
model+training	0.641

The average semantic coverage score was found to be 0.653, indicating that over 60% of generated questions were matched effectively using metadata alone, without relying on paper full text. This result demonstrates the utility of paper attributes in estimating reviewer focus and enables metadata-only retrieval of relevant reviewer questions. Among the best covering combinations, data type+model (0.715) and data type+evaluation metric (0.697) yielded the highest coverage, suggesting that reviewers often connect methodological aspects with the type of data being used as stated in Table 6. Similarly, combinations involving dataset (e.g., dataset+model at 0.671 and dataset+data type at 0.669) also produced consistently high coverage, highlighting the importance of dataset-related information in guiding reviewer attention. In contrast, combinations such as data type+training (0.592) or training+evaluation metric (0.615) resulted in lower coverage, indicating that training details alone are less predictive of reviewer questioning patterns.

4.2.2 Question Matching Evaluation Against Extracted Review Questions

The set of generated reference questions was used to further validate the question matching pipeline. For each paper, retrieved questions were compared with generated questions using

cosine similarity, and coverage was computed as the proportion of matched questions exceeding the 0.6 similarity threshold.

The number of qwen2.5_m-q_v2-extracted review questions from all groundtruth reviews per paper ranged from 0 to 23. In parallel, metadata-guided matched questions varied in size, often exceeding 100 candidates depending on the number of matched metadata entries. After filtering based on semantic similarity and clustering, 12 representative questions per paper were selected. These covered an average of 65% of generated questions, demonstrating both coverage and diversity in the final question set.

4.2.3 Review Generation and Evaluation Against Ground Truth Reviews

To assess the quality of automatically generated reviews, I designed a pipeline that takes an unseen paper's full text, abstract, and matched reviewer questions as input, and generates a review.

I employed `Qwen2.5-14B` through the ChatOllama interface to generate a critical review. The input prompt includes the paper abstract, full content, and a list of retrieved reviewer questions (from previous metadata-based matching). The model is instructed to only output answers to applicable questions, maintaining a professional and objective tone. The prompt is formatted and passed using LangChain's [49] `ChatPromptTemplate` and `Runnable` pipeline to modularize the prompt construction and inference pipeline, simplifying experimentation and scaling.

Review quality evaluation was assessed by comparing model-generated reviews to ground truth reviews written by human reviewers for ICLR 2020 papers in the dataset. When multiple reference reviews were available, they were concatenated to form a single comprehensive review per paper.

Model Configurations. Three different model configurations were evaluated to compare review generation quality:

- **Baseline 1 – Qwen3(qwen3_b1_v1):** A general instruction was given to generate an ICLR-style review using only the full paper content, without access to matched or extracted questions.
- **Baseline 2 – Qwen2.5 without given questions(qwen2.5_b2_v1):** The same instruction and input setup as Baseline 1 were used, but with a different underlying model.
- **Proposed Method – Qwen2.5 with given review questions(qwen2.5_m-q_v2):** The same base model was used, but with an additional input of 12 matched reviewer questions per paper.

Evaluation Criteria. The evaluation pipeline was applied calculating ROUGE-1[50], ROUGE-2[50], ROUGE-L[50], BERTScore[48], and cosine similarity between Sentence-BERT embeddings[46] of the generated and reference reviews.

- **ROUGE-1:** Measures unigram (single word) overlap between the generated text and the reference, capturing basic lexical similarity.
- **ROUGE-2:** Measures bigram (two-word sequence) overlap, reflecting the fluency and preservation of local word order.
- **ROUGE-L:** Measures the longest common subsequence (LCS) between the generated text and the reference, capturing fluency and sentence-level structural similarity.
- **BERTScore:** Measures token-level similarity based on contextualized embeddings from a pre-trained language model.
- **Cosine Similarity:** Computes the average cosine distance between Sentence-BERT embeddings of the generated and ground truth reviews.

These metrics focus on capturing deeper semantic alignment beyond surface-level lexical overlap, allowing a more meaningful comparison of review content quality.

4.2.4 Review Generation Evaluation

Review-Level Similarity Results. For ICLR 2020 papers, baseline model reviews and question-guided model reviews are generated and compared with the papers' ground truth aggregated reviews. Table 7 reports performance across models using review-wise metrics. Since the ROUGE scores are low across all the model variants, it does not give a reliable indication of content similarity, highlighting the importance of semantic-oriented metrics like BERTScore and cosine similarity for evaluating review generation. The qwen2.5_m-q_v2 guided by questions achieved the highest BERTScore (0.803), reflecting deeper semantic relevance. In contrast, the unguided qwen2.5_b2_v1 model outperformed in cosine similarity, indicating stronger surface-level alignment. The baseline qwen3_b1_v1 lagged behind both variants across all metrics, except ROUGE-2 with quite low values.

Table 7: Review-wise Similarity Scores Across Model Variants

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	Cosine Sim
qwen3_b1_v1	0.326	**0.084**	0.145	0.589	0.672
qwen2.5_b2_v1	**0.379**	0.079	0.138	0.580	**0.683**
qwen2.5_m-q_v2 + Questions	0.237	0.034	**0.217**	**0.803**	0.477

'b1', 'b2': Baseline variants using Qwen-3 and Qwen-2.5, respectively
'm-q': Main model Qwen-2.5 with reviewer questions
'v1': without question guidance 'v2': with question guidance

Limitations of Review-Level Evaluation. Initial evaluation at the review level was found to be insufficient due to inconsistencies in length and verbosity across generated and reference reviews. For example, a generated review for a paper includes 9 sentences while its ground truth aggregated review includes 32 sentences. This often distorted similarity measurements and misrepresented content alignment. Moreover, although both the generated and ground truth reviews contain sentences related to aspects such as summary, motivation, and originality,

which are not the focus of this study, the evaluation specifically concentrated on the aspects of clarity, comparison, and replicability. To address these challenges, a sentence-level evaluation strategy was adopted, where each generated sentence was aligned with its most semantically similar sentence in the ground truth review.

Sentence-Level Evaluation Methodology. Two complementary methods were used to evaluate semantic alignment at the sentence level: (1) cosine similarity based on sentence embeddings, and (2) BERTScore F1 based on token-level contextual similarity. Sentence embeddings were computed using the `all-MiniLM-L6-v2` model. For both similarity metrics, a pairwise similarity matrix was constructed between n generated and m reference sentences. The matrix was padded to size $\max(n,m) \times \max(n,m)$ with very low constants to handle unequal lengths since the number of sentences could differ across the two sets. The Hungarian algorithm[51] was then applied on the negative of this matrix, yielding an optimal one-to-one alignment of sentences. This procedure aimed that each generated sentence was aligned with at most one reference sentence, and vice versa, while maximizing the total similarity score used for reporting average BERTScore and cosine similarity values.

Sentence-Level Similarity Results. Table 8 presents the sentence-level evaluation. qwen3_b1_v1 achieved the highest average scores (BERTScore: 0.868, cosine: 0.702), followed closely by qwen2.5_b2_v1 without questions. The qwen2.5_m-q_v2 with questions yielded slightly lower average similarity but generated a larger number of high-quality alignments, reflecting broader topic coverage.

Table 8: Sentence-wise Similarity Scores Across Model Variants

Model ID	Avg BERTScore	Avg Cosine Similarity
qwen3_b1_v1	**0.868**	**0.702**
qwen2.5_b2_v1	0.866	0.701
qwen2.5_m-q_v2	0.860	0.685

Aspect-Based Filtering. Motivated by the need to control for decision-relevant aspects, the review components that are known to artificially inflate sentence-level similarity, namely *summary, motivation, soundness, originality, and substance* were removed. This choice is consistent with preprocessing strategies in question generation, where filtering is applied to reduce bias introduced in reviews. For this, I used the fine-tuned aspect tagger introduced in [38], which was originally used to construct the ASAP-Review dataset. All golden reviews in the test set, as well as LLM-generated reviews (both question-guided and unguided), were tagged with aspect labels. Sentences associated with the excluded aspects (along with their positive and negative sentiment variants) were removed from the texts. The remaining sentences were then reevaluated under the same sentence-wise comparison protocol. These sentences tagged with *comparison, clarity, replicability*. As expected, this exclusion led to a decrease in the observed average similarity, since content breadth was reduced and the reviews were specific to experiments or clarity issues. The results of this filtered evaluation are shown in Table 9.

Table 9: Sentence-wise Similarity Scores After Filtering Aspects Across Model Variants

Model ID	Avg BERTScore	Avg Cosine Similarity
qwen3_b1_v1	**0.8794**	**0.3542**
qwen2.5_b2_v1	0.8644	0.3515
qwen2.5_m-q_v2	0.8685	0.3023

Table 10: Examples of sentence pairs aligned by the Hungarian algorithm for qwen2.5_m-q_v2 alongside their matched BERT scores

Generated Sentence	Reference Sentence	BERT F1
Although the model outperforms several baselines, it would be beneficial to include comparisons with other recent layerwise sampling methods that also address over-smoothing and sparsity issues (e.g., DropEdge, Jodie).	Experimental with Jumping Neural Network is missing to understand how the proposed solution improves over existing solutions for over-smoothing.	0.8556
The paper could provide a more detailed ablation study to better understand the contributions of each component in BLS-GAN.	It would be helpful to even couple it with Fast-GCN/AS-GCN sampler to better understand the benefits of this paper.	0.8667
Could you provide more detailed explanations or pseudocode to clarify how the bi-diffusion sampling process works?	Illustrations would also be very helpful to provide an intuition about formulas 4, 5, and 6.	0.8448
Can you conduct an ablation study to analyze the impact of each component in BLS-GAN on model performance and efficiency?	It is currently impossible to disentangle the effects of the two parts of BLS-GAN.	0.8888
To replicate your experiments: - Ensure all datasets used are clearly specified.	Some of the terminologies used are not clearly described and are not explained prior to the usage.	0.8525

In Table 10, aligned sentence pairs produced by the Hungarian algorithm for the question-guided qwen2.5_m-q_v2 model, along with their BERTScore similarity are shown. The similarity values indicating strong semantic correspondence despite lexical variation. These high-scoring pairs cover common review themes such as missing baselines, ablation studies, replicability details, and methodological clarifications, and they directly contribute to the aggregated similarity metrics reported in Table 8 and Table 9.

Interpretation of Sentence-Level Trends. Despite lower average similarity scores, the Qwen2.5 model with questions produced more sentence pairs above the similarity threshold. This suggests that question-based prompting enhances content diversity and coverage while slightly reducing local similarity due to broader topical scope. Unguided models offered stronger lexical alignment per sentence, but lacked the structural grounding provided by targeted questions. After aspect-based filtering, a reduction in overall similarity averages was observed, consistent with the expectation that removing core argumentative aspects reduces direct overlap between generated and reference reviews.

4.3 Qualitative Study

In addition to to the quantitative experiments, I conducted a qualitative analysis to evaluate the effectiveness of the proposed question-matching framework on recent study papers, including the generated questions aligned with major guideline questions, and compared the performance of an automated reviewing tool with the result obtained using the proposed methodology. The goal of this study was to assess whether the system could produce paper-specific questions and reviews that align with actual reviewer concerns and comments.

To evaluate proposed question-matching framework on recent study papers, I made use of a new feature introduced in ICLR 2025: the *Reviewer Questions* section besides strengths and weaknesses instead of freely written review comments, which includes questions explicitly written by reviewers to express their specific doubts or expectations. This provides reviewers with more structured guidance, enabling them to ask more targeted questions about specific aspects of the paper, such as the methodology or experimental setup. I collected a total of 100 papers from ICLR 2025 via the OpenReview platform to be used in this analysis. The selection was performed in an ad-hoc manner with stratified sampling based on acceptance (50 accepted - 50 rejected). It also reflects the range of questions reviewers posed under the new structured guidelines.

To illustrate the generated questions aligned with major guideline questions, I used the 2025 review guidelines of three major conferences: NeurIPS, ICLR, and ACL. This allowed for a comparison between the automatically generated questions and established evaluation criteria, ensuring that the proposed question-matching framework captures the key aspects emphasized by expert reviewers.

To compare the features of reviews of a paper from the ICLR 2020 proceedings (specifically ICLR_2020_395), reviews are collected from four sources: ground-truth human reviews, the proposed methodology, and two automated tools, OpenReviewer [52] and ACL Ready [2]. This comparison aims to evaluate how well automated and pipeline-generated reviews capture the key aspects of the paper relative to human reviews, summarized in Table 14.

4.3.1 Data Preprocessing

Non-experimental sections, including *References* and *Related Work*, were excluded to ensure that the content used in the analysis reflects the core contributions and methodology of each paper. This filtering procedure is consistent with the preprocessing strategy applied in earlier stages of the pipeline.

To limit the scope of the study to applied research, *Qwen 2.5-14b*, the same LLM utilized in the quantitative study, was used to classify the remaining papers based on their content. Each paper, with non-essential sections removed, was input into the LLM for classification as theoretical or non-theoretical. The model identified 79 out of 100 papers as non-theoretical. Only these papers were retained for subsequent metadata extraction and qualitative evaluation.

4.3.2 Metadata Extraction

Text segments corresponding to five predefined metadata categories were extracted from each of the selected papers. These categories include *model type*, *training strategy*, *evaluation metrics*, *dataset*, and *methodology*. Extraction was performed using prompt-based methods designed and described in earlier sections of the thesis.

In metadata generation task, one of the best-extracted features by the LLM was *dataset* information, as most papers contained explicit descriptions of the datasets used, enabling accurate extraction of dataset names. Another feature extracted properly was *model* information, possibly supported by the inclusion of a taxonomy list of model names. In contrast, the *training strategy* feature showed the lowest performance. The main difference between the highly- and less-extracted features lies in the retrieval task: for highly extracted features with less NA values as shown in Table 12, the system could directly extract names from the text, whereas for the low-performing feature, it needed to interpret the content and infer the relevant information, requiring semantic understanding beyond simple extraction (See Table 11). This NA values also could indicate a clarity issue in papers since the metadata relevant paragraphs/features are not available to be extracted.

4.3.3 Qualitative Comparison of Golden and Matched Questions

In qualitative analysis, Table 11 presents a sampled set of golden and matched questions along with extracted metadata of the papers. In entries where metadata fields such as dataset, model, evaluation metric, and data type are provided, the matched questions clearly reflect the perspective of the golden questions. For example, the paper with "div3, end0" dataset and GPT-nano and LSTM, the golden and matched questions show similarity in perspective: both sets ask about model behavior (number of models trained, stability), quantitative evaluation (effects of cluster denoising, metrics for structured outputs), and interpretability (understanding model components). The golden question "How many models were trained on each dataset? How stable are the results across different training runs?" is matched with "Why were the data sources chosen, and how do their limitations affect result validity?", showing alignment in perspective regarding *model setup and experimental stability*. In the IMDB-BINARY, COLLAB, and MUTAG datasets with AGSN+MLP and GSN+MLP, the golden and matched questions similarly reflect runtime, adaptivity, and methodological evaluation. The matched questions reframe the golden questions in slightly different terms, but maintain the same evaluative perspective on performance and method comparison. For Transformer-based models, even though some metadata fields are NA, the golden and matched questions share a perspective on understanding model functionality. For TabM and TabMmini models, both golden and matched questions focus on performance assessment, feature representation, and methodological choices.

In entries where metadata is partially missing (e.g., training strategy or evaluation metric marked NA), the similarity in perspective is still observable but less precise. For example, for a Transformer-based model, the golden question "The notation is a bit unclear... This makes it difficult to understand exactly how the method works" is matched with "Could you provide detailed explanations or examples of how your model components work compared to standard models?", reflecting related intent (clarification of method), but with broader focus.

Across all sampled rows, matched questions frequently reflect the thematic areas of the golden questions, such as runtime, dataset, or methodological clarity, even when metadata is missing. However, rows with complete metadata show closer alignment in perspective, as illustrated by multiple paired questions.

Table 11: Qualitative comparison of overlapping golden and matched questions

Dataset	Model	Training Strategy	Evaluation Metric	Data Type	Golden Questions	Matched Questions
div3, end0	GPT-nano, LSTM	NA	valid rate,cross-entropy loss	text data	How many models were trained on each dataset? How stable are the results across different training runs? Did you perform any quantitative tests on the effects of cluster denoising? What would you say is the primary goal of the paper? Is it about interpretability. . . ?	Why were the data sources chosen, and how do their limitations affect result validity? Which metrics demonstrate that structuring [output data feature] enhances [sub task] performance? Could you clarify how the proposed [model component] simplifies understanding of [concept] in [domain]?
IMDB-BINARY, COL-LAB, MUTAG	AGSN+MLP, GSN+MLP	Supervised Learning	validation accuracy, mean, std	Other	What's the running time compared with other GNN methods, Could you provide runtime comparisons with baselines? In what scenarios does adaptivity help/hurt performance? why we need adaptivity? What are the advantages of the proposed adaptive wavelet diffusion compared to other methods [1-2]?	How does the proposed method compare to existing methods in terms of effectiveness and efficiency? What specific limitations do sections have, and how does your new model overcome them? How does the proposed method compare to existing methods in terms of effectiveness and efficiency?
NA	Transformer-based model	NA	NA	NA	The notation is a bit unclear... This makes it difficult to understand exactly how the method works. Can you provide specific runtime comparisons between Contrast-CAT and other methods...? Attributing at token-level is not enough for interpretability...	Could you provide detailed explanations or examples of how your model components work compared to standard models? How does the proposed method compare to other state-of-the-art models in terms of performance? How does a rich model component affect the exploration policy in proposed methods?
NA	TabM, Tab-Mmini	NA	NA	Other	Could the authors provide more detailed diversity metrics for the predictions of the 32 base models? How does TabM determine the representation of each sample...? Why is k not included in the hyper-parameter search for TabM?	How do the performance measures differ between the proposed and baseline methods? How are output features generated from input data? What is the rationale for excluding specific features in the preprocessed data?

Table 12: Number of NA values for each metadata among 79 non-theorethical papers

Metadata Type	Number of NA values
Data Type	14
Dataset	20
Model	12
Training Strategy	33
Evaluation Metric	18

Table 12 reports the number of missing metadata values across 79 non-theoretical papers. The highest number of NA(Not Applicable) values occurs in Training Strategy (33 instances), followed by Dataset (20) and Evaluation Metric (18). Data Type and Model show fewer missing values, with 14 and 12 instances respectively. These missing labels typically arise when the LLM cannot assign a metadata tag due to insufficient context or the presence of datasets and models that are not part of its prior knowledge.

4.3.4 Question Set Diversification and Comparison

Each 2025 paper was matched to papers from the ICLR 2019 dataset by computing similarity over their extracted metadata representations. Reviewer questions associated with the most similar 2019 papers were retrieved to form a candidate question set corresponding to each 2025 paper.

A subset of questions was selected from the candidate pool to maximize aspectual diversity and minimize redundancy. The resulting set was treated as the *metadata-matched questions* associated with each target paper.

To assess the quality and relevance of the matched questions, a comparison was conducted against the reviewer-provided questions included in the ICLR 2025 submissions. The evaluation criteria focused on thematic overlap, contextual relevance, and coverage across metadata aspects. This comparison served as a qualitative assessment of the framework's ability to retrieve questions that are both meaningful and consistent with those posed by human reviewers.

4.3.5 Reviewer Guidelines from Major Conferences vs Proposed Framework

The qualitative analysis also aimed to compare the coverage of extracted review questions from past reviews with the reviewer guideline questions from major conferences in the deep learning domain. To illustrate the similarities in question diversity and coverage, Table 13 presents a sampled comparison along with the thematic aspects of the questions.

The extracted questions cover multiple thematic areas, including evaluation metrics, model configuration, clarity, reproducibility, experimental design, computational efficiency, and data selection. Questions on reproducibility, such as "How do you implement an objective function with one, two, or three parameters to enhance replicability?", correspond to guidelines requesting detailed training specifications. Questions on computational resources and memory usage, for instance "How do the proposed methods differ from previous approaches in addressing high memory usage?", align with guidelines requesting hardware and compute details. Framework questions often provide more concrete details, such as specific parameter counts or model components, compared to broader guideline questions.

Questions on model interpretation, like "What do [model components] represent?" and "How does [model component] differ from existing [baseline method] in terms of parameter efficiency and differentiability?", match guidelines addressing paper objectives and novelty. Questions on clarity, such as "Could you explain how your method ensures clarity in adapting and combining approaches for readers unfamiliar with these methods?", align with guidelines on clarity, correctness, and evaluation. Experimental design and reproducibility questions, including

"How do experiment setups reflect abstract ideas?", correspond to guidelines requiring clearly specified training details and empirical support.

Questions on comparative performance and efficiency, for example "How do model components estimate uncertainty compared to LSTMs or GRUs?" and "What improvements does your method offer over existing approaches in task efficiency without runtime fine-tuning?", align with guidelines emphasizing novelty, reproducibility, and evaluation rigor. Questions on data selection, such as "Are specific metrics or thresholds used in selecting training data?", match guidelines requesting data statistics. Overall, the framework reflects reviewer concerns captured in major conference guidelines while providing detailed, task-specific focus.

The extracted review questions provide coverage consistent with major conference reviewer guidelines. They reflect the same thematic areas and reviewer concerns while capturing specific methodological details relevant for evaluation. Although the questions are similar to guideline questions, each question is not applicable to every paper. By matching the questions to specific papers, this framework identifies the questions that are relevant for each paper, highlighting an advantage of this approach over using generic guideline questions.

Table 13: Matched new proposed framework questions with guideline questions and thematic aspects

Extracted Question from Past Reviews	**Matching Guideline Question (Source)**	**Aspect/Theme**
How do the performance measures differ between the proposed and baseline methods?	Are the claims supported (empirically or theoretically)? (ICLR)	Evaluation metrics
How do model component choices affect performance, and what are the optimal parameter values?	Is the hyperparameter search space and setup detailed? (ACL(Association for Computational Linguistics))	Model configuration
What challenges or limitations exist in replicability regarding data, training, and testing setups?	Are models/datasets reproducible or verifiable through described steps? (NeurIPS)	Reproducibility
What criteria were used to evaluate the proposed methods, and how do they differ from the compared methods in terms of metrics and experimental settings?	Are result statistics (error bars, means, etc.) included? (ACL)	Evaluation criteria
Are the selected data subsets' predictive performance and generalization comparable to those of the full dataset?	Are data stats (splits, counts) reported? (ACL)	Data evaluation
Why were certain features excluded from the preprocessed data?	Are data stats (splits, counts) reported? (ACL)	Data preprocessing

Continued on next page

Table 13 – continued from previous page

Extracted Question from Past Reviews	Matching Guideline Question (Source)	Aspect/Theme
Can you compare the performance metrics of the proposed [model component] with previous state-of-the-art models on the [task], focusing on accuracy improvements without using pre-trained contextual encoders?	Are the results novel, reproducible, and rigorous? (ICLR)	Comparative performance
What challenges or limitations exist in replicability regarding data, training, and testing setups?	Are training details (e.g., splits, hyperparameters) clearly specified? (NeurIPS)	Reproducibility
How does the paper ensure understanding of model components for readers with different technical backgrounds?	Are there any weak points in clarity, correctness, or evaluation? (ICLR)	Clarity/Accessibility
What do [model components] represent?	What is the objective of the paper? (ICLR)	Model interpretation
How does [model component] differ from existing [baseline method] in terms of parameter efficiency and differentiability?	Are the results novel, reproducible, and rigorous? (ICLR)	Comparative performance
Could you explain how your method ensures clarity in adapting and combining approaches for readers unfamiliar with these methods?	Are there any weak points in clarity, correctness, or evaluation? (ICLR)	Clarity/Accessibility
How does the proposed method compare to other unsupervised learning methods in terms of performance and applicability for the task?	Are the results novel, reproducible, and rigorous? (ICLR)	Comparative performance
How does the proposed framework differ from other non-linear dynamical systems models?	Are the results novel, reproducible, and rigorous? (ICLR)	Comparative performance
How do you implement an objective function with one, two, or three parameters to enhance replicability?	Are training details (e.g., splits, hyperparameters) clearly specified? (NeurIPS)	Reproducibility
What improvements does your method offer over existing approaches in task efficiency without runtime fine-tuning?	Are the results novel, reproducible, and rigorous? (ICLR)	Efficiency / Performance

Continued on next page

Table 13 – continued from previous page

Extracted Question from Past Reviews	Matching Guideline Question (Source)	Aspect/Theme
How do experiment setups reflect abstract ideas?	Are the claims supported (empirically or theoretically)? (ICLR)	Experimental design
What metrics were used for quantitative evaluation, and how do they compare to other studies?	Are result statistics (error bars, means, etc.) included? (ACL)	Evaluation metrics
How does weight initialization affect model performance compared to random initialization?	Is the hyperparameter search space and setup detailed? (ACL)	Model configuration
How do you choose regularization methods, and how do they affect alignment quality across domains?	Is the hyperparameter search space and setup detailed? (ACL)	Model configuration
Why was only the loss function used for comparison?	Are result statistics (error bars, means, etc.) included? (ACL)	Evaluation criteria
How do model components estimate uncertainty compared to LSTMs or GRUs?	Are the results novel, reproducible, and rigorous? (ICLR)	Comparative performance
How do the proposed methods differ from previous approaches in addressing high memory usage, and what specific modifications were made?	Are compute details (hardware, time, memory) disclosed? (NeurIPS)	Computational efficiency
Are specific metrics or thresholds used in selecting training data?	Are data stats (splits, counts) reported? (ACL)	Data selection
What computational resources are needed to train model components, and how do they compare to training the full model with the entire dataset?	Are model parameters, compute resources, and budget shared? (ACL)	Computational resources

4.3.6 Comparison of Ground Truth and Generated Reviews

To demonstrate the comparison between two automated reviewing tools, OpenReviewer [52] and ACL Ready [2], and the reviews obtained using the proposed methodology and ground-truth human reviews. OpenReviewer is an open-source tool based on the Llama-OpenReviewer-8B model and generates structured feedback following a review template consistent with the ICLR 2025 review format. ACL Ready is another open-source AI assistant tool that adapts the ACL Responsible NLP Research Checklist to help authors and reviewers, based on the Meta-Llama-3.1-8B-Instruct-Turbo model. For comparison, the ground-truth human reviews, the reviews generated by the proposed methodology, OpenReviewer, and ACL Ready are collected for a specific paper from the ICLR 2020 proceedings. The paper under examination, with ICLR_2020_395 paper_id, addresses task-free continual learning,

where task boundaries are unknown during training. It proposes a novel deep learning model, which adaptively expands a set of neural network experts to handle new data while retaining previously learned knowledge. This expansion-based approach grounded in different tasks including image classification and image generation. The comparison of the four different reviews is summarized in Table 14.

Table 14: Review Comparison for ICLR_2020_395

Review Type	**Review Content**
Ground Truth (Human Reviewer)	The paper proposes an elegant method for task-free continual learning problems. It has nicely pointed out that the conventional continual learning algorithms had the limitation of knowing the task boundaries. Followings are my. By applying DPM, the authors proposed a method of automatically determining whether to add a new expert for a new task or train the existing experts. While the Dirichlet Process Mixture(DPM) is not new, applying such nonparametric method to continual learning is new. The experimental results are impressive given the single-epoch setting. Pros 1. Good experimental results for the task-free setting, in which no information about task boundaries is given . Particularly, even with smaller memory-usage than the experience replay (ER) methods, the proposed method achieves better results.2.Many past work should suffer from increased number of tasks due to the model capacity limit, but the proposed method efficiently expands the model capacity . 3. Writing flow is good and is easy to follow. Cons & Questions 1. I am not sure whether the proposed methods should work well for ‘ all cases. Is there any cases in which the proposed DPM would fail? 2. What happens when you actually know the task boundaries? Would following the framework with known number of experts also excel other methods? 3. Can you apply this to the RL setting?

Continued on next page

Table 14 (continued)

Review Type	Review Content
Proposed Method	The document provides a detailed comparison between CN-DPM and CURL in terms of their generative processes and approaches to continual learning. However, there are several areas where the clarity and comprehensiveness could be improved for better understanding and replicability: 1. **Clarity on Key Concepts**: The introduction should clearly define key concepts such as "generative replay" and how it differs from other methods like CN-DPM's approach to prevent catastrophic forgetting. This would help readers understand these distinctions by page one. 2. **Comparison with Previous Studies**: It is crucial to compare the experimental protocols used in this study with those of previous studies on continual learning, especially focusing on how generative replay and model expansion are implemented differently. This comparison can enhance replicability and provide a clearer context for evaluating the contributions of CN-DPM. 3. **Interaction Between Model Components**: The document should elaborate more on how different components of CN-DPM interact with each other to achieve its objectives. For instance, explaining how the generative process interacts with the clustering mechanism would be beneficial. [...] 5. **Performance and Applicability Comparison**: While the document discusses how CN-DPM compares to CURL, it should also include a broader comparison with other continual learning methods regarding performance metrics such as accuracy retention over time and computational efficiency. 6. **Improvements for Querying States**: The paper could discuss in more detail what improvements CN-DPM offers specifically when querying states from input data compared to existing approaches. 7. **Steps Ensuring Replicability**: It would be helpful to outline the specific steps taken to ensure replicability of the methods described, especially regarding training setups and hyperparameter tuning. 8. **Exclusion of Certain Methods**: The rationale for excluding certain continual learning methods from comparison should be explained more clearly. Including these methods could provide a fuller validation of CN-DPM's performance relative to the state-of-the-art. [...] 12. **Metrics Used for Evaluation**: The document should clearly state which metrics were used to evaluate model objectives and why these particular metrics were chosen over others. By addressing these points, the document would provide a clearer understanding of CN-DPM's contributions to continual learning while also facilitating its replicability and broader applicability in the field.

Continued on next page

Table 14 (continued)

Review Type	Review Content
OpenReviewer	Summary: This paper proposes a new method for task-free continual learning based on Dirichlet process mixture models. The proposed model learns a mixture of experts, where each expert is a discriminative model (classifier) and a generative model (VAE). The mixture weights are determined by a Dirichlet process prior, which allows for an adaptive number of experts. The method is evaluated on a variety of datasets and compared to several baselines. Soundness: 3 Presentation: 3 Contribution: 3 Strengths: * The paper is well written and easy to follow. * The proposed method is novel and well motivated. * The method is evaluated on a variety of datasets and compared to several baselines. Weaknesses: * The method is only compared to a few baselines. In particular, it would be interesting to see how the method compares to other methods that use a mixture of experts, such as the ones cited in the paper (Expert gate, Progress & compress,...). * The proposed method uses a Dirichlet process prior, which is a nonparametric method for determining the number of experts. However, the number of experts is eventually bounded (as described in Appendix H). It would be interesting to see how the method performs with a parametric method for determining the number of experts, such as a Gaussian process prior. Questions: * How does the method compare to other methods that use a mixture of experts? * How does the method perform with a parametric method for determining the number of experts, such as a Gaussian process prior? Flag For Ethics Review: No ethics review needed. Details Of Ethics Concerns Rating: 8 Confidence: 4

Continued on next page

Table 14 (continued)

Review Type	Review Content
ACL Ready	A1: Did you discuss the limitations of your work? Response: 5 Conclusion. The conclusion section includes a reflection on the limitations of the work. A2: Did you discuss any potential risks of your work? Response: 5 Conclusion. The conclusion section is a common place to discuss potential risks and ... [...] B1: Did you cite the creators of artifacts you used? Response: 4 Experiments. The section '4 Experiments' details the experiments conducted using artifacts created by others, such as datasets and models. B2: Did you discuss the license or terms for use and/or distribution of any artifacts?Response: 4 Experiments. The section '4 Experiments' discusses the experimental details and model architectures... [...] B4: Did you discuss the steps taken to check whether the data that was collected/used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it? Response: None. The information regarding checking for offensive content and identifiers, as well as steps taken to protect/anonymize data, is not explicitly mentioned in the provided context. [...] C1: Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used? Response: C Model Architectures and Experimental Details. The section 'C Model Architectures and Experimental Details' typically includes details about the model parameters, architecture, and experimental setup, which would likely cover reporting the number of parameters in the models used. [...] C3: Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run? Response: 4 Experiments. The section '4 Experiments' provides detailed experimental results, including the average of 10 runs, and discusses the performance of the model across different scenarios. C4: If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?Response: C Model Architectures and Experimental Details. The section 'C Model Architectures and Experimental Details' provides detailed information... [...]

When comparing the four reviews shown in Table 14, their different emphases highlight how review sources prioritize aspects that ultimately shape acceptance decisions. The human review comments on the method's novelty in applying Dirichlet process mixtures to task-free continual learning and emphasizes strong results and efficiency, while raising open questions about limitations and applicability in reinforcement learning or known task boundaries. The human review provides comments that do not cover all aspects in a structured way including all the guiding aspects. They also include comments that may not strongly influence the acceptance decision, such as those on soundness or summary. However, since the reviewers are experts, their comments are generally more factual and detailed compared to other generated reviews, providing insights on figures, tables, and other areas where LLMs struggle to make accurate inferences.

OpenReviewer review emphasizes novelty, empirical soundness, and overall contribution, providing detailed summaries, highlighting strengths, identifying weaknesses, and raising questions about comparisons or methodological choices. For example, OpenReviewer notes the novelty of the mixture-of-experts method, discusses evaluation across multiple datasets, and explicitly raises questions about alternative expert selection strategies. This reviewer type captures subtle methodological nuances and broader scientific context, making the feedback highly informative but still not fully structured across all aspects.

ACL Ready, in contrast, enforces checklist-based compliance. Its review systematically audits whether the paper addresses limitations, potential risks, proper citation and licensing of data artifacts, anonymization and ethical safeguards, and reporting of computational budgets and model parameters in specific sections companied by section name. For instance, ACL Ready review notes coverage of conclusion limitations, experimental details for artifact usage, and reporting of descriptive statistics and computational setup, while pointing out missing information regarding anonymization or offensive content checks. This approach is less evaluative of scientific novelty, but is highly effective in detecting gaps in transparency, reproducibility, and ethics. It also retrieves the relevant section name to the checklist question.

The proposed pipeline review prioritizes decision-critical aspects like replicability, clarity, and comprehensive methodological evaluation based on the guiding questions retrieved. In the review text provided, the proposed method highlights several key areas for improvement: clarifying core concepts such as generative replay, comparing experimental protocols and performance with previous studies, elaborating on interactions between model components, explaining rationale for method exclusions, and specifying metrics and steps for ensuring replicability. This review type emphasizes structural completeness and actionable guidance, focusing on aspects that directly affect reproducibility and evaluation, complementing the expert judgment of human reviewers and the checklist enforcement of ACL Ready.

4.4 Discussion

The proposed methodology has several potential practical applications. First, it can assist conference and journal editors by providing a systematic and transparent way to track reviewer intent and coverage. Second, authors can use the generated question sets to guide manuscript

revisions, focusing on aspects frequently highlighted in past reviews. Third, junior reviewers may benefit from prior question patterns as a training aid, improving consistency in evaluations.

The study's findings reveal a trade-off between semantic similarity and structural quality in LLM-generated reviews. Although the framework does not verify whether matched questions are answered in the paper content, the review generation experiments show that including such questions still affects semantic alignment with ground truth reviews. Question-guided qwen2.5_m-q_v2 achieved the highest BERTScore (0.803), while unguided qwen2.5_b2_v1 had higher cosine similarity (0.683), and qwen3_b1_v1 left behind on most metrics. ROUGE scores were low across all models, indicating limited lexical overlap. These results suggest that question-guided input improves semantic relevance and focus on critical paper aspects, even if surface-level similarity is lower. Combined with metadata coverage analysis, where higher coverage corresponds to fewer missing values in Data Type and Model, the findings indicate that completeness of metadata and inclusion of targeted questions are key factors for effective review generation (See Table 7).

An important consideration in baseline selection was computational efficiency and reproducibility. Although DeepSeek-R1 achieved results comparable to Qwen models, it was excluded due to its reliance on reasoning steps and the need for additional post-processing. Consistent with the pilot study setup, the choice favored models that offer strong performance while being computationally less demanding, operable under limited local resources, and configurable to produce deterministic outputs without extended reasoning. This design decision ensured that baseline models not only represent competitive semantic performance but also reflect realistic deployment constraints for large-scale review generation.

The similarity in model scores does not imply that the underlying reasoning is equivalent. This proposed question-guided approach improves clarity, control, and openness by explicitly aligning review content with specific questions. This method supports Explainable AI (XAI) by grounding the review generation process with paper-specific questions, showing what aspects were addressed. While baseline models are fluent, this method provides peer reviews with paper-relevant concerns, offering both semantic quality and structural clarity.

Since review-level scores may obscure sentence-level variation, I additionally align individual sentences to capture fine-grained semantic correspondence. Sentence-level evaluation revealed that review-level metrics alone were insufficient due to differences in length and verbosity between generated and reference reviews. Aligning individual sentences using the Hungarian algorithm allowed a more precise measurement of semantic correspondence. qwen3_b1_v1 and qwen2.5_b2_v1 achieved slightly higher average BERTScore (0.868 and 0.866) and cosine similarity (0.702 and 0.701), while the question-guided qwen2.5_m-q_v2 scored 0.860 and 0.685, respectively. Despite lower averages, the guided model produced a larger number of high-quality alignments, covering key aspects such as clarity, comparison, and replicability. Aspect-based filtering, which excluded summary, motivation, and originality sentences, reduced overall similarity but highlighted decision-relevant content; the question-guided model maintained competitive BERTScore (0.8685) despite a decrease in cosine similarity. Examples of aligned sentence pairs show strong semantic alignment on common review points such as missing baselines, ablation studies, and replicability details. These results indicate that question-guided input increases coverage of relevant content and

ensures structural alignment across important review aspects, even if local lexical similarity is slightly reduced (See Table 10).

Beyond semantic similarity, the completeness of paper metadata also plays a central role in question coverage. The analysis of missing metadata values provides further insight into the observed coverage patterns. Metadata types with fewer missing values, such as Model (12 NA) and Data Type (14 NA), correspond to higher coverage in question matching, indicating that successfully extracted metadata directly contributes to retrieving reviewer questions. In contrast, metadata with more missing entries—particularly Training Strategy (33 NA) and Dataset (20 NA) show lower coverage, suggesting that gaps in metadata extraction limit the framework's ability to identify corresponding questions. These observations reinforce that the effectiveness of metadata-guided question matching is closely tied to the completeness of available paper attributes, and highlight the importance of metadata extraction as a prerequisite for generating comprehensive and representative reviewer question sets (See Table 12).

In the qualitative study, the observed overlap between generated and golden questions suggests the framework captures the intent and structure of expert reviewer inquiry rather than relying on lexical matches. The match of questions extended across diverse categories such as computational efficiency, clarity, and comparative performance which indicates the adaptability of questions to varied research focuses.

The qualitative analysis shows that the proposed framework can generate and match questions that align with reviewer concerns and conference guidelines. When complete metadata is available, matched questions closely mirror golden reviewer questions in both perspective and specificity, while missing metadata reduces precision but still preserves thematic overlap. NA counts indicate that Training Strategy and Dataset are the most frequent sources of missing labels, limiting question coverage (See Table 12 and Table 6). Comparisons with ICLR 2025 reviewer questions confirm that the framework retrieves paper-specific, decision-relevant questions rather than generic prompts (See Table 11).

Alignment with NeurIPS, ICLR, and ACL guidelines demonstrates that the framework captures established review dimensions such as clarity, reproducibility, and evaluation aspects, while offering the additional benefit of guiding questions to individual papers (See Table 13).

To contextualize these findings, the proposed framework's outputs against human and automated reviews are compared. In the analysis of comparison of reviews, each review type offers complementary insights. Human reviewers provide nuanced and factual feedback, though coverage across all aspects is inconsistent. OpenReviewer emphasizes novelty and methodological depth, raising key questions but lacking structured replicability checks. ACL Ready enforces checklist compliance, ensuring transparency, ethics, and reproducibility; for the items it flags as missing in the paper, the proposed method review addresses these gaps by systematically evaluating replicability, clarity, and methodological aspects. These perspectives support more robust acceptance decisions.

These findings above highlight complementary strengths, but one limitation remains in the current framework. The framework does not verify whether a matched question is answered in the paper content but does identify likely relevant concerns based on metadata. This enables generalization across domains. Despite this limitation, the framework offers practical

applications for different stakeholders in peer review. Future work may combine metadata-based prompting with content-aware reasoning for more comprehensive review support with also peer review automating tools already in use.

CHAPTER 5

CONCLUSION

5.1 Conclusion

The aim of this study is to develop a structured, metadata-guided framework for extracting, generalizing, and aligning reviewer questions to new scientific submissions. By actual reviewer comments and structured paper features such as model, datasets, data type, training strategy, and evaluation metrics, the framework enables the generation of context-aware, reusable, and explainable reviewer questions. This approach seeks to improve the relevance, consistency, and explainability of LLM-generated peer reviews across diverse research domains. To achieve this, I developed a methodology that aims two strategies: metadata-based question extraction and paper-specific question matching for review generation.

A pilot study was conducted on 45 ICLR 2017 papers to construct and test prompts for reviewer question extraction, ensuring coverage of applied deep learning studies with available data. Prompt trials showed that template-based and few-shot approaches yielded the most consistent and structured outputs for question extraction, generalization, and metadata generation. The method generalizes extracted review questions by replacing domain-specific terms with standardized entity labels, enabling reusable template of questions. Entity labels were iteratively created through semantic grouping and group validation, distinguishing between related but contextually different terms. This process was applied to questions from 138 reviews among 45 papers, resulting in a final entity list of 54 representative labels for systematic question generalization. Reviewer questions for test set were retrieved by matching test set paper metadata embeddings with previously reviewed papers using FAISS-based nearest neighbor search. Multiple metadata field combinations were used to increase coverage, and similarity scores above 0.5 determined relevant matches. To reduce redundancy, matched questions were clustered with K-means, and the centroid question from each cluster was selected as representative for review generation.

The quantitative results show that metadata-guided question matching achieved an average semantic coverage of 0.653, indicating that over 60% of extracted review questions were effectively retrieved using paper metadata alone. Across papers, 12 representative questions selected via clustering covered on average 65% of the extracted review questions of test papers. These results demonstrate that metadata embeddings can approximate reviewer focus while maintaining diversity in the selected questions. Representative question selection via clustering covers 65% of extracted review questions while reducing redundancy. Each combination

coverage ratio varied. The higher coverage scores may stem from two potential factors: (i) such features might indeed be the most frequently referenced by reviewers, or (ii) as also noted in the qualitative analysis, these are the metadata types that the LLM was most successful at extracting and tagging, thereby inflating their apparent coverage performance.612

Question-guided review generation produces broader coverage and high-quality sentence alignments despite slightly lower average similarity. Aspect-based filtering confirms that question-guided outputs effectively target critical review dimensions about experiment, methodology, and clarity. Matched questions align with official guidelines, ensuring domain-specific relevance and reusable templates of questions based on past review metadata alignment.

Sentence-level evaluation complemented review-wise metrics by aligning each generated sentence with its most semantically similar reference sentence using cosine similarity and BERTScore. While Qwen2.5 guided by questions scored slightly lower on average similarity (BERTScore: 0.860, cosine: 0.685) than unguided variants, it produced a larger number of high-quality alignments, indicating broader topic coverage and more informative content. This demonstrates that question-guided review generation effectively balances semantic alignment with coverage across key review aspects. Aspect-based filtering removed sentences associated with summary, motivation, soundness, originality, and substance to focus on decision-relevant aspects such as comparison, clarity, and replicability. After filtering, Qwen2.5 guided by questions still produced numerous high-quality sentence alignments with strong semantic similarity (BERTScore up to 0.8888), indicating effective coverage of key review content despite reduced overall similarity scores. This demonstrates that question-guided generation improves content diversity and alignment with critical review aspects while maintaining meaningful semantic correspondence.

Qualitative study indicates that many matched questions exhibited thematic and conceptual overlap with official reviewer prompts and established conference guidelines, particularly in evaluation metrics, reproducibility, and model configuration. Besides, the similarity in perspective between golden and matched questions is stronger when metadata is clearly extracted, and broader or less precise when metadata fields are missing. The table entries directly illustrate this pattern through side-by-side comparison of the sampled questions. The extracted review questions align with major conference guidelines but are matched to specific papers, ensuring that only relevant questions are applied to each paper. This selective matching highlights the framework's advantage over using generic guideline questions. As key findings indicate that average of 60% of extracted questions are covered by matched questions using paper metadata alone.

5.2 Limitations

This study has several constraints that may affect generalization and scalability. The pilot study used 45 papers from ICLR 2017, which may not capture the full diversity of writing styles, methodologies, or topics across conference years or domains. The stopping criterion was based on saturation in extracted aspects and metadata patterns; a larger sample might reveal additional nuances. The entity list was built from the same 45 papers and 138 reviews, which may limit coverage of domain-specific terminology in broader datasets. The metadata

schema excludes an optimization field, potentially reducing the ability to match papers with similar optimization strategies despite related questions being present. BERT-based similarity was employed for validating generated reviews and matched question coverage, providing alignment but adding computational overhead as the LLM evaluations require computational resource and time. The current implementation is partially manual, which limits scalability and reproducibility. The reliance on LLMs and structured metadata also makes the outputs sensitive to model quality and data completeness. The coverage ratio used to evaluate question matching is computed in an ad-hoc manner, which restricts the generalization of the analysis in large-scale or real-time scenarios. This heuristic-based calculation may not fully capture all relevant matches or the nuanced alignment between generated and reference questions, limiting the precision of the evaluation. Embedding-based similarity alone may also miss nuanced reasoning required for review generation. The question clustering process includes all generated questions, regardless of relevance or answerability, which may reduce the utility of the final question set. The pipeline covering labeling, extraction, and review generation relies heavily on large language models. Intermediate outputs were not fully verified, and the error rate or its downstream impact remains unknown.

5.3 Future Work

Future research could expand the dataset to multiple years and conferences to improve representativeness and reduce sampling bias. Entity extraction should be applied to a larger and more diverse set of papers and reviews to capture a wider range of domain-specific concepts. Adding an optimization metadata field may improve retrieval by capturing methodological similarities absent in the current schema. Combining embedding-based similarity with reasoning or prompt-based guidance could reduce computational costs and improve reasoning coverage. Filtering out generic or already answered questions, and generating unanswered but contextually expected ones (e.g., checklist-style prompts), could enhance review and question quality. Quantifying LLM error rates in intermediate outputs would clarify their effect on downstream performance and guide quality assurance measures. The coverage ratio could be computed using more principled or alternative metrics, such as probabilistic matching, weighted similarity thresholds, or semantic alignment scores, to provide a more robust and accurate evaluation of question matching. In practical use as shown in Appendix A.1, automating metadata extraction, question generalization, and matching steps would improve efficiency. Furthermore, integrating a user-friendly interface or dashboard could enable researchers to interact with the question pool, filter or prioritize questions, and apply the method to new submissions more effectively.

REFERENCES

[1] D. Scherbakov, N. Hubig, V. Jansari, A. Bakumenko, and L. A. Lenert, "The emergence of large language models as tools in literature reviews: a large language model-assisted systematic review," *Journal of the American Medical Informatics Association*, vol. 32, p. 1071–1086, May 2025.

[2] M. Galarnyk, R. Routu, K. Bheda, P. Mehta, A. Shah, and S. Chava, "Acl ready: Rag based assistant for the acl checklist," 2024.

[3] Q. Wang, Q. Zeng, L. Huang, K. Knight, H. Ji, and N. F. Rajani, "Reviewrobot: Explainable paper review generation based on knowledge synthesis," *arXiv preprint arXiv:2010.06119*, 2020.

[4] T. Batane, "Turning to turnitin to fight plagiarism among university students," *Journal of Educational Technology & Society*, vol. 13, no. 2, pp. 1–12, 2010.

[5] Grammarly Inc., "Grammarly." `https://www.grammarly.com`, 2024. Accessed: August 13, 2025.

[6] J. H. and, "Writing overleaf—collaboration to inspire tomorrow's researchers," *Editorial Office News*, vol. 9, pp. 22–23, oct 2016.

[7] A. Checco, L. Bracciale, P. Loreti, S. Pinfield, and G. Bianchi, "Ai-assisted peer review," *Humanities and social sciences communications*, vol. 8, no. 1, pp. 1–11, 2021.

[8] K. Kousha and M. Thelwall, "Artificial intelligence to support publishing and peer review: A summary and review," *Learned Publishing*, vol. 37, no. 1, pp. 4–12, 2024.

[9] S. Basuki and M. Tsuchiya, "The quality assist: A technology-assisted peer review based on citation functions to predict the paper quality," *IEEE Access*, vol. 10, pp. 126815–126831, 2022.

[10] J. Lin, J. Song, Z. Zhou, Y. Chen, and X. Shi, "Automated scholarly paper review: Concepts, technologies, and challenges," *Information fusion*, vol. 98, p. 101830, 2023.

[11] J. Lin, J. Song, Z. Zhou, Y. Chen, and X. Shi, "Moprd: A multidisciplinary open peer review dataset," *Neural Computing and Applications*, vol. 35, no. 34, pp. 24191–24206, 2023.

[12] K. Wang and X. Wan, "Sentiment analysis of peer review texts for scholarly papers," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 175–184, 2018.

[13] J. Chen, W. Su, Z. Chu, H. Li, Q. Ai, Y. Liu, M. Zhang, and S. Ma, "An automatic and cost-efficient peer-review framework for language generation evaluation," 2024.

[14] Y. Jin, Q. Zhao, Y. Wang, H. Chen, K. Zhu, Y. Xiao, and J. Wang, "Agentreview: Exploring peer review dynamics with llm agents," 2024.

[15] R. Alqifari, "Question answering systems approaches and challenges," in *Proceedings of the Student Research Workshop Associated with RANLP 2019*, pp. 69–75, 2019.

[16] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," 2016.

[17] E. Fortier-Dubois and D. Rosati, "Using contradictions improves question answering systems," *arXiv preprint arXiv:2211.05598*, 2022.

[18] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: a large-scale dataset for fact extraction and verification," *arXiv preprint arXiv:1803.05355*, 2018.

[19] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, eds.), (Brussels, Belgium), pp. 2369–2380, Association for Computational Linguistics, Oct.-Nov. 2018.

[20] Y. Zhang, K. Liu, S. He, G. Ji, Z. Liu, H. Wu, and J. Zhao, "Question answering over knowledge base with neural attention combining global knowledge information," 2016.

[21] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," 2017.

[22] R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi, "Swag: A large-scale adversarial dataset for grounded commonsense inference," 2018.

[23] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman, "Newsqa: A machine comprehension dataset," 2017.

[24] A. Rogers, M. Gardner, and I. Augenstein, "Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension," *ACM Computing Surveys*, vol. 55, p. 1–45, Feb. 2023.

[25] S. Filice, G. Horowitz, D. Carmel, Z. Karnin, L. Lewin-Eytan, and Y. Maarek, "Generating diverse q&a benchmarks for rag evaluation with datamorgana," *arXiv preprint arXiv:2501.12789*, 2025.

[26] R. Zhang, J. Guo, L. Chen, Y. Fan, and X. Cheng, "A review on question generation from natural language text," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 1, pp. 1–43, 2021.

[27] Y. Zhang, X. Liu, Y. Sun, A. Alharbi, H. Alzahrani, B. Alomair, and D. Song, "Can llms design good questions based on context?," *arXiv preprint arXiv:2501.03491*, 2025.

[28] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, D. R. Krathwohl, *et al.*, *Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain*. Longman New York, 1956.

[29] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," *arXiv preprint arXiv:1705.00106*, 2017.

[30] R. Y. Pang, A. Parrish, N. Joshi, N. Nangia, J. Phang, A. Chen, V. Padmakumar, J. Ma, J. Thompson, H. He, and S. R. Bowman, "Quality: Question answering with long input texts, yes!," 2022.

[31] Z. Li, M. Cukurova, and S. Bulathwela, "A novel approach to scalable and automatic topic-controlled question generation in education," in *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pp. 148–158, 2025.

[32] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

[33] K. Ronanki, B. Cabrero-Daniel, and C. Berger, "Prompt smells: An omen for undesirable generative ai outputs," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, pp. 286–287, 2024.

[34] K. Ronanki, S. Arvidsson, and J. Axell, "Prompt engineering guidelines for using large language models in requirements engineering," *arXiv preprint arXiv:2507.03405*, 2025.

[35] Y. Cheng, J. Chen, Q. Huang, Z. Xing, X. Xu, and Q. Lu, "Prompt sapper: A llm-empowered production tool for building ai chains," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 5, pp. 1–24, 2024.

[36] Z. Tan, D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, M. Karami, J. Li, L. Cheng, and H. Liu, "Large language models for data annotation and synthesis: A survey," 2024.

[37] S. Yadav, T. Choppa, and D. Schlechtweg, "Towards automating text annotation: A case study on semantic proximity annotation using gpt-4," 2024.

[38] W. Yuan, P. Liu, and G. Neubig, "Can we automate scientific reviewing?," *CoRR*, vol. abs/2102.00176, 2021.

[39] Allenai, "Allenai/science-parse: Science parse parses scientific papers (in pdf form) and returns them in structured form.."

[40] W. Yuan and P. Liu, "Kid-review: Knowledge-guided scientific review generation with oracle pre-training," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 11639–11647, Jun. 2022.

[41] L. Rimkus, J. Verbickas, and R. Batista-Navarro, "Reviewer 2 must be stopped: Transformer-based approaches for predicting paper acceptance," in *Information Management and Big Data* (J. A. Lossio-Ventura, E. Ceh-Varela, G. Vargas-Solar, R. Marcacini, C. Tadonki, H. Calvo, and H. Alatrista-Salas, eds.), (Cham), pp. 249–260, Springer Nature Switzerland, 2024.

[42] S. Kumar, H. Arora, T. Ghosal, and A. Ekbal, "Deepaspeer: towards an aspect-level sentiment controllable framework for decision prediction from academic peer reviews," in *Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries*, JCDL '22, (New York, NY, USA), Association for Computing Machinery, 2022.

[43] P. Muangkammuen, F. Fukumoto, J. Li, and Y. Suzuki, "Intermediate-task transfer learning for peer review score prediction," in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 40–47, 2023.

[44] S. Singh, M. Singh, and P. Goyal, "COMPARE: A taxonomy and dataset of comparison discussions in peer reviews," *CoRR*, vol. abs/2108.04366, 2021.

[45] IEEE.

[46] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.

[47] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The faiss library," 2024.

[48] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," 2020.

[49] LangChain, "Langchain documentation." `https://www.langchain.com`, 2025. Accessed: 2025-08-15.

[50] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[51] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[52] M. Idahl and Z. Ahmadi, "Openreviewer: A specialized large language model for generating critical scientific paper reviews," 2025.

APPENDIX A

EXTRA MATERIAL

A.1 Practical Guideline for Applying the Methodology

Phase 1 – Preprocessing

1. **Metadata Related Text Extraction (Past Submissions)**
   - **Input:** Previous submissions (JSON, CSV files)
   - **Tool:** LLM
   - **Output:** Relevant metadata text
2. **Metadata Extraction (Past Submissions)**
   - **Input:** Extracted metadata-related text
   - **Tool:** LLM
   - **Output:** Structured metadata (Paper ID, dataset, data type, model, training strategy, evaluation metric)
3. **Entity List Construction (Pilot Study)**
   - **Input:** Sample reviewer comments and generated questions from these comments
   - **Tool:** Human annotators
   - **Output:** Entity name list for question generalization
4. **Reviewer Question Extraction**
   - **Input:** Review texts
   - **Tool:** LLM
   - **Output:** Raw reviewer questions
5. **Question Generalization**
   - **Input:** Raw reviewer questions + entity name list
   - **Tool:** LLM
   - **Output:** Generalized questions

Phase 2 – Review Generation

1. **Metadata Extraction (New Submission)**

    - **Input:** New submission metadata
    - **Tool:** LLM
    - **Output:** Structured metadata

2. **Question Retrieval & Clustering**

    - **Input:** New paper metadata + past question pool
    - **Tool:** Cosine embedding similarity, K-means clustering
    - **Output:** Clustered set of relevant questions

3. **Review Generation with Curated Questions**

    - **Input:** Paper-specific set of clustered questions
    - **Tool:** LLM
    - **Output:** Question-guided, domain-specific draft review

A.2 Prompt Trials

A.2.1 Primary Question Extraction Prompt Example

Prompt (Instruction)

"You are given review text. Extract semantically meaningful units that reflect **clarity, replicability, or comparison**. Then, convert each into a direct, relevant question.
Strict Instructions:
- Focus only on content discussing clarity, reproducibility, or comparison of methods, results, or ideas.
- Do **not** generate questions based on summaries, motivation, background, or general overview.
- Do **not** generate questions involving:

- Figures (e.g., Figure 1, visuals, diagrams)
- Tables (e.g., Table 2, numeric comparisons)
- Equations or math symbols
- Specific page numbers or section numbers

- Do **not** include questions that refer to:

- The reviewer (e.g., "reviewer suggests. . . ")
- The paper itself (e.g., "Does the paper. . . ")
- Any subjective evaluation or judgment

Keep questions **short, precise, and content-focused**.
Format your output **exactly** like this, with no extra text or explanation:
Q1: [Question]
A1: [Answer from the review text]
Q2: [Question]
A2: [Answer from the review text] ..."

Input Context

This paper presents a set of experiments ...
...
Please state that random labels came from the same set as the true labels, to clarify the experiment.

Intended Response

Q1: Is the comparison done between state of model/theory and the paper's contribution?
A1: These observations are used to argue that classical theory ...

Q2: What is the advantage of this paper comparing to the state of art model/theory?
A2: The classical generalization theory is indeed often at a loss ...

Q3: Is the experimental setup consistent and stated clearly?
A3: A minor clarification comment ...

Actual Model Output

Q1: How can the experiment be clarified ...?
A1: On page 2, ... random labels came from the same set ...

A.2.2 Final Question Generation

"""Here is the abstract of the paper to help you understand the context: abstract

Here is the review text for generating questions: "review"

Extract concise, content-based questions that a reviewer might ask based on this review. Focus only on clarity, comparison, replicability aspects—skip visuals, tables, and equations.

- Format your output like this:

Q1: [Question] Q2: [Question] ... """

A.2.3 Final Question Generalization

Question Prompt:

You are given a list of generated questions related to a scientific paper questions. Your task is to generalize each question via an entity list so that it is applicable to other similar papers. Follow the instructions strictly. Use only the provided entity list entitiesto replace domain-specific terms with square-bracketed general entities (e.g., [model component], [dataset], etc.).

Example: Original: """Q1: Why is recursion considered important for neural programming architectures?""" Generalized: """GQ1: Why is [concept] considered important for [domain]?"""

Return only the following format, without any additional explanation or extra text: GQ1:[Generalized Question 1] GQ2:[Generalized Question 2] """

A.2.4 Question Post Processing Chain

Question prompt for each chain:
Chain 1:
"""Given questions: question
and entity list:
entities,
do two things:

1. Replace any bracketed term not in the entity list with the closest fitting one from the list (keep square brackets). 2. If bracketed entities are repeated (e.g., "[section] and [section]"), replace with plural or higher-level form (e.g., "[sections]", "[document]")."""

Return only the revised questions. No explanation or extra text. """
Chain 2:
"""From the following questions:
question
and considering the list of general entity names:
entities,
check if the questions are **short, clear, concise, and understandable. Remove unnecessary adjectives redundant phrases, or repeated entity references to reduce length while keeping the main point intact."""
Chain 3:
"""From the following questions:
question
and considering the list of general entity names:
entities,
exclude any questions that contain references to 'Figure', 'Table', 'Equation', 'Section', 'Visual', or numbered references (e.g., Fig. 2, Eq. (3)).Return only the questions without any explanation or comment."""

A.2.5 Evaluation Metric Text Extraction from Paper Content

Question Prompt:
"""From the following questions:text,
check only the paragraphs including information about the evaluation methods or metrics used for validation of experiment results. Return only the found related text as one piece without any explanation or comment """

A.2.6 Evaluation Metric Extraction

Question Prompt:

"""Here is a scientific paper text:"text".
Extract the evaluation metric utilized in the paper based on the given part of the text to evaluate the proposed approach. Provide only the list of metrics. If there is no given text, respond as "NA". Provide the output with no additional explanations, or commentary.

Return the result in the following format:

[Answer 1], [Answer 2], ...

"""

A.2.7 Dataset Text Extraction from Paper Content

Question Prompt:

"""From the following questions:text,
check only the paragraphs including information about the datasets being used in the experiment or method sections. Return only the found related text as one piece without any explanation or comment """

A.2.8 Dataset Extraction

Question Prompt:

"""Here is a scientific paper text: "text".
Extract the precise information about datasets used for either training or evaluation of proposed methods based on the given part of the text. Provide only the list of datasets and which data type it is from the given data type list. Provide the output with no additional text, explanations, or commentary. Return the result in the following format:

Dataset 1: [Answer], Data type 1: [Answer], Dataset 2: [Answer], Data type 2: [Answer], ..

"""

A.2.9 Data Type Extraction

Question Prompt:

"""Here is a scientific paper text: "text".
Extract the information about data type used either in training or in evaluation based on the given part of the text. Choose the proper data type either as text, image, or other. If there is no given text, respond as "NA". Provide the output with no additional explanations, or commentary.

"""

A.2.10 Training Strategy Text Extraction from Paper Content

"""From the following questions:text,
check only the paragraphs including information about training strategy being used in experiment or methodology sections. Return only the found related text as one piece without any explanation or comment """

A.2.11 Training Strategy Extraction

Question Prompt:

"""Here is a scientific paper text: "text".
Extract the information about training procedure used for models based on the given part of the text. Choose the proper training style from well-known training styles such as adverserial training, novel training, semi-supervised training, curriculum learning, federated learning, unsupervised learning, and etc.

If there is no given paper text, respond as "NA". Provide the output with no additional explanations, or commentary.

Return the result in the following format:

[Answer]

"""

A.2.12 Model Text Extraction from Paper Content

Question Prompt:

"""From the following questions:text, check only the paragraphs including information about the models, architectures, methods used in the experiment section. Return only the found related text as one piece without any explanation or comment """

A.2.13 Model Type Extraction

Question Prompt:

"""Here is a scientific paper text: "text". Extract the models used in the paper. Provide only the models with no additional text, explanations, or commentary.

Return the result in the following format:

[Answer 1] [Answer 2] .. """

A.3 July 2025 IEEE Taxonomy

Taxonomies = ["Artificial intelligence", "Affective computing", "AI accelerators", "Artificial general intelligence", "Autonomous robots", "Bio-inspired computing", "Cognitive systems", "Commonsense reasoning", "Context awareness", "Context management", "Conversational artificial intelligence", "Chatbots", "Virtual assistants", "Cooperative systems", "Edge AI", "Explainable AI", "Foundation models", "Generative AI", "Prompt engineering", "Intelligent systems", "Autonomous systems", "Collective intelligence", "Hyper-intelligent systems", "Intelligent automation", "Intelligent robots", "Knowledge based systems", "Expert systems", "Mobile agents", "Knowledge engineering", "Inference mechanisms", "Knowledge acquisition", "Knowledge discovery", "Knowledge representation", "Large language models", "Generative Pre-trainer transformer", "Learning (artificial intelligence)", "Distance learning", "Naive Bayes methods", "Nearest neighbor methods", "Learning systems", "Active learning", "Backpropagation", "Cognitive systems", "Collective learning", "Electronic learning", "Hybrid learning", "Immersive learning", "Learning automata", "Learning management systems", "Self-supervised learning", "Semisupervised learning", "Stability plasticity", "Supervised learning", "Unsupervised learning", "Weak supervision", "Machine learning", "Adversarial machine learning", "Association rule learning", "Attention mechanisms", "Automated machine learning", "Boosting", "Concept drift", "Deep learning", "Deep reinforcement learning", "Diffusion models", "Dimensionality reduction", "Ensemble learning", "Explainable AI", "Federated learning", "Few shot learning", "Generative Pre-trainer transformer", "High dimensional data", "Hyperparameter optimization", "Imitation learning", "Incremental learning", "Multi-armed bandit problem", "One shot learning", "Random forests", "Reinforcement learning", "Relevance vector machines", "Representation learning", "Robot learning", "Statistical learning", "Tiny machine learning", "Transfer learning", "Machine listening", "Prediction methods", "Linear predictive coding", "Predictive coding", "Predictive encoding", "Predictive models", "Retrieval augmented generation", "Text to image", "Text to video", "Virtual artifact", "Autonomous mental development", "Computational intelligence", "Computation theory", "Computational chemistry", "Computational complexity", "Computational statistics", "Concurrent computing", "Greedy algorithms", "Reservoir computing", "Support vector machines", "Turing completeness", "Computational cybernetics", "Evolutionary computation", "Evolutionary robotics", "Particle swarm optimization", "Fuzzy systems", "Fuzzy control", "Fuzzy neural networks", "Hybrid intelligent systems", "Genetic algorithms", "Genetic operators", "Logic", "Fuzzy logic", "Takagi-Sugeno model", "Multivalued logic", "Probabilistic logic", "Sufficient conditions", "Machine intelligence", "Pattern analysis", "Neural networks", "Artificial neural networks", "Autoencoders", "Convolutional neural networks", "Hebbian theory", "Long short term memory", "Neural architecture search", "Residual neural networks", "Self-organizing feature maps", "Spiking neural networks", "Biological neural networks", "Cellular neural networks", "Feedforward neural networks", "Extreme learning machines", "Multilayer perceptrons", "Graph neural networks", "Graph convolutional networks", "Multi-layer neural network", "Neural network compression", "Neural network hardware", "Perception evolution networks", "Radial basis function networks", "Recurrent neural networks", "Bidirectional long short term memory", "Echo state networks", "Hopfield neural networks", "Stability plasticity"]

A.4 Final Entity List

Entities = ["optimization method", "regularization methods", "training strategy", "experiment setup", "training setup", "testing setup", "comparative method", "baseline method", "core model architecture", "model component ", "algorithm", "evaluation metric", "main method", "main approach", "theoretical concept", "task definition", "context", "field", "document", "reference", "technique", "experiment setup", "evaluation", "metric", "comparison metric", "alternative metric", "named metric", "data/dataset", "meta data", "training data", "data collection", "rule", "test data", "validation data", "mathematical concept", "implementation", "process", "literature", "reference paper", "model feature", "evaluation property", "model aspect", "concept", "results", "output", "predicted output", "evaluation output", "input data feature", "output data feature", "hyperparameter", 'parameter', "parameter settings", "parameters choices", "system"]

A.5 Pilot Study

Accepted Papers (ICLR 2017)

ICLR_2017_1: Making Neural Programming Architectures Generalize via Recursion

ICLR_2017_2: End-to-end Optimized Image Compression

ICLR_2017_3: Optimization as a Model for Few-Shot Learning

ICLR_2017_4: Learning End-to-End Goal-Oriented Dialog

ICLR_2017_5: Towards Principled Methods for Training Generative Adversarial Networks

ICLR_2017_6: Reinforcement Learning with Unsupervised Auxiliary Tasks

ICLR_2017_7: Multi-Agent Cooperation and the Emergence of (Natural) Language

ICLR_2017_8: Understanding deep learning requires rethinking generalization

ICLR_2017_9: Neural Architecture Search with Reinforcement Learning

ICLR_2017_10: Q-Prop: Sample-Efficient Policy Gradient with An Off-Policy Critic

ICLR_2017_11: Learning to Act by Predicting the Future

ICLR_2017_12: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima

ICLR_2017_13: Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data

ICLR_2017_14: Amortised MAP Inference for Image Super-resolution

ICLR_2017_15: Learning Graphical State Transitions

ICLR_2017_16: Maximum Entropy Flow Networks

ICLR_2017_17: Topology and Geometry of Half-Rectified Network Optimization

ICLR_2017_18: Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer

ICLR_2017_19: Learning Visual Servoing with Deep Features and Fitted Q-Iteration

ICLR_2017_20: Stochastic Neural Networks for Hierarchical Reinforcement Learning

ICLR_2017_21: Nonparametric Neural Networks

ICLR_2017_22: Distributed Second-Order Optimization using Kronecker-Factored Approximations

ICLR_2017_23: Pruning Filters for Efficient ConvNets

ICLR_2017_24: Learning to Generate Samples from Noise through Infusion Training

ICLR_2017_25: Filter Shaping for Convolutional Neural Networks

ICLR_2017_26: Normalizing the Normalizers: Comparing and Extending Network Normalization Schemes

ICLR_2017_27: Multilayer Recurrent Network Models of Primate Retinal Ganglion Cell Responses

ICLR_2017_28: Improving Generative Adversarial Networks with Denoising Feature Matching

ICLR_2017_29: Efficient Vector Representation for Documents through Corruption

ICLR_2017_30: Learning Invariant Feature Spaces to Transfer Skills with Reinforcement Learning

Rejected Papers (ICLR 2017)

ICLR_2017_300: Neural Code Completion

ICLR_2017_301: Here's My Point: Argumentation Mining with Pointer Networks

ICLR_2017_302: Tree-Structured Variational Autoencoder

ICLR_2017_303: Learning Approximate Distribution-Sensitive Data Structures

ICLR_2017_304: Memory-augmented Attention Modelling for Videos

ICLR_2017_305: Taming the Waves: Sine as Activation Function in Deep Neural Networks

ICLR_2017_306: Convolutional Neural Networks Generalization Utilizing the Data Graph Structure

ICLR_2017_307: Evaluation of Defensive Methods for DNNs against Multiple Adversarial Evasion Models

ICLR_2017_308: Collaborative Deep Embedding via Dual Networks

ICLR_2017_309: Transformation-based Models of Video Sequences

ICLR_2017_310: Neural Causal Regularization under the Independence of Mechanisms Assumption

ICLR_2017_311: Recurrent Coevolutionary Feature Embedding Processes for Recommendation

ICLR_2017_312: L-SR1: A Second Order Optimization Method for Deep Learning

ICLR_2017_313: Inverse Problems in Computer Vision using Adversarial Imagination Priors

ICLR_2017_314: Leveraging Asynchronicity in Gradient Descent for Scalable Deep Learning