CitAI Research Centre

School of Mathematics, Computer Science and Engineering

City St George's, University of London

Semantic Fidelity in Specialized Domains: Advancing Language Models through Adaptive Learning, Collective Reasoning, and Consensus Evaluation

Author:
Sami Saadaoui

Academic Supervisors:
Prof. Eduardo Alonso
Dr. Daniel Chicharro Raventos
Dr Simone Krummaker

A thesis submitted in partial fulfillment of the requirement for the degree of Doctor of Philosophy

March 2026

To my father, whose endless curiosity and thirst for
knowledge inspired me to pursue this path.
To my mother, whose unconditional love and belief in
me have always lifted me up.
To my loving wife, whose constant support and
unwavering faith have driven me forward.
And to my wonderful siblings, whose encouragement and
belief in me have always been a source of strength.

Declaration of Original and Sole Authorship

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

I grant powers of discretion to the City St George's, University of London librarian to allow the dissertation to be copied in whole or in part without further reference to myself (the author). This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgment.

Signature:
[redacted]
Sami Saadaoui
Date: March 2026

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Eduardo Alonso, for guidance, patience, and consistent support throughout the course of this research. The advice and insight provided during this work have been invaluable and have shaped my development as an independent researcher. The constructive feedback and thoughtful discussions have strengthened both this thesis and my understanding of the field.

I am also deeply thankful to Dr Esther Mondragón for her generous support during my studies.

I would like to thank my colleagues at CitAI for their collaboration, and the stimulating research environment they provided. Their support has made this journey more rewarding and enjoyable.

Finally, I would like to extend my heartfelt appreciation to my family. Their support, understanding, and belief in me have sustained and motivated me through every stage of this journey. They have been a constant source of strength, and I am truly grateful for their presence in my life.

Contents

List of Figures

List of Tables

Listings

Abstract

The effective deployment of language models in specialized domains such as finance and medicine requires addressing three coupled challenges: learning domain-specific representations, generating semantically faithful and comprehensive outputs, and evaluating quality without extensive human annotation. This dissertation addresses these challenges through the unifying concept of **semantic fidelity**, defined as the preservation of intended meaning and relations among domain concepts across (i) representation, (ii) generation, and (iii) evaluation. This work develops three complementary contributions spanning representation learning, multi-agent reasoning, and consensus-based evaluation.

First, *Adaptive Masked Language Modeling (AMLM; Chapter 3)* introduces a domain adaptation approach that dynamically prioritizes domain-specific terminology during pre-training through adaptive importance weighting. By incorporating multiple signals of token importance together with stabilization mechanisms, AMLM ensures robust training under highly skewed importance distributions. Evaluated on financial-domain tasks, AMLM demonstrates improvements in semantic textual similarity while producing more compact and semantically coherent representations. These results suggest that adaptive importance weighting provides an effective, architecture-agnostic path to domain specialization.

Second, *Collective Intentional Reading through Reflection and Refinement (CIR3; Chapter 4)* introduces a multi-agent framework for generating question–answer pairs that are both comprehensive and faithful to technical context. CIR3 applies collective intelligence principles through structured coordination that balances perspectival diversity with semantic alignment to the source material. Agents iteratively refine outputs through interaction protocols that prevent premature consensus. Experiments across financial and medical datasets demonstrate substantial improvements in both comprehensiveness and faithfulness over strong baselines, while reducing duplication and over-specificity.

Third, the *consensus-based evaluation framework (Chapter 5)* enables rigorous assessment without reliance on human-annotated gold standards. The framework establishes semantic consensus among multiple models and quantifies inter-model reliability through hierarchical clustering across multiple semantic granularities. Target systems are evaluated using agreement metrics that balance fine-grained and holistic semantic alignment. Cross-domain validation in finance and medicine demonstrates strong consensus reliability, robust system alignment, and stability across both general-purpose and domain-informed embeddings, suggesting that multi-model consensus offers a practical alternative to annotation-intensive evaluation.

Together, these contributions enhance semantic fidelity in specialized domains. AMLM learns domain-aware representations through weighted loss functions; CIR3 structures collective reasoning to produce faithful, comprehensive outputs; and the consensus framework provides a principled means of evaluation without gold standards. The components are modular and interoperable, supporting independent use or integration into a unified pipeline for high-stakes NLP applications where semantic precision is critical.

The findings demonstrate that (i) training-objective design can enable efficient domain specialization without architectural changes, (ii) collective intelligence mechanisms can balance diversity and convergence for reliable reasoning, and (iii) multi-model consensus with quantified reliability offers a practical alternative to annotation-intensive evaluation. Collectively, these results outline a coherent methodological framework for improving language model fidelity in specialized domains.

Abbreviations

AE	Answer Extraction
QA	Question Answering
ODQA	Open-Domain Question Answering
DL	Deep Learning
NLP	Natural Language Processing
NLU	Natural Language Understanding
NLI	Natural Language Inference
KB	Knowledge Base
IR	Information Retrieval
MRC	Machine Reading Comprehension
STS	Semantic Textual Similarity
MLM	Masked Language Modeling
NSP	Next Sentence Prediction
PLM	Pre-trained Language Model
RAG	Retrieval-Augmented Generation
DPR	Dense Passage Retrieval
NER	Named Entity Recognition
POS	Part-Of-Speech
TF-IDF	Term Frequency–Inverse Document Frequency
KD	Knowledge Distillation
CP	Continued Pre-training
DAPT	Domain-Adaptive Pre-training
TAPT	Task-Adaptive Pre-training
PEFT	Parameter-Efficient Fine-Tuning
ICL	In-Context Learning

CoT	Chain-of-Thought
ToT	Tree of Thoughts
QG	Question Generation
QAG	Question-Answer Generation
LM	Language Model
LLM	Large Language Model
LLM-MAS	LLM-based Multi-Agent Systems
AMLM	Adaptive Masked Language Modeling
ESS	Effective Sample Size
kNN-MLE	k-Nearest Neighbors Maximum Likelihood Estimator
R@K	Recall@K
MRR	Mean Reciprocal Rank
MAP	Mean Average Precision
nDCG	Normalized Discounted Cumulative Gain
Soft-F1	Soft token overlap F1
Bip-F1	Bipartite (Hungarian) F1
DB	Davies–Bouldin Index
CH	Calinski–Harabasz Score
MSE	Mean Squared Error
MAE	Mean Absolute Error
CI	Confidence Interval
IRR	Inter-Rater Reliability
MoE	Mixture of Experts
TMS	Transactive Memory Systems
C3	Collective Cognitive Convergence
CI	Collective Intelligence
CIR3	**C**ollective **I**ntentional **R**eading through **R**eflection and **R**efinement

Chapter 1

Introduction

1.1 Motivation and Context

Advances in language models have enabled systems to perform complex tasks in generation, representation learning, and semantic understanding [1, 2]. However, specialized domains (e.g., finance, medicine, and law, which involve complex terminology and knowledge structures) continue to pose challenges for **semantic fidelity**, which we define as the degree to which a system preserves domain-specific meanings and conceptual relations across three interconnected stages: (i) representation, capturing domain semantics within model representations; (ii) generation, producing outputs that are comprehensive and semantically faithful to source content; and (iii) evaluation, assessing semantic equivalence despite lexical variation.

Specialized domains present three interconnected challenges that distinguish them from general language use. First, they exhibit *high semantic density*: specialized terminology carries precise technical meanings, and subtle distinctions between related concepts can have significant practical implications. For instance, in investment, the terms “index fund”, “exchange-traded fund (ETF)”, and “mutual fund” all refer to pooled investment vehicles, yet differ fundamentally in their operational mechanism, typical application, and broader economic impact. Second, these domains involve *implicit knowledge structures*, where expert understanding depends not only on explicit statements but also on inferring unstated relationships, assumptions, and domain conventions. For example, in medical diagnosis, the relationship between symptoms and conditions often hinges on clinical context and epidemiological patterns not explicitly articulated in individual case reports. Third, specialized domains face *evaluation scarcity*: the cost and expertise required to produce high-quality annotations create bottlenecks in system development and validation. This scarcity is further compounded by the high semantic density and implicit knowledge, requiring evaluators to possess deep domain expertise to assess whether systems accurately capture subtle conceptual distinctions and unstated relationships.

These challenges manifest across the full lifecycle of language model deployment. During pre-training and adaptation, standard language modeling objectives weight all prediction errors uniformly, failing to prioritize domain-specific terminology that carries disproportionate semantic weight. For downstream tasks such as question generation, summarization, or information extraction, models must not only recall relevant information but also reason over implicit relationships and produce outputs that are both comprehensive and faithful to the source material. During evaluation, the absence of large-scale annotated datasets creates a validation gap, leading to reliance on proxy metrics with limited diagnostic value or costly human evaluation.

This dissertation addresses these challenges through an integrated framework structured across three stages: (i) learning mechanisms that guide models to internalize domain-specific semantics, (ii) reasoning architectures that leverage collective intelligence for complex generation, and (iii) evaluation methodologies that assess semantic quality without requiring extensive human annotation. Chapters 3–5 instantiate these elements through AMLM (representation), CIR3 (reasoning), and a consensus-based framework (evaluation).

1.2 Research Challenges and Gaps

The challenges of semantic fidelity in specialized domains intersect with several active research areas in Natural Language Processing (NLP), including domain adaptation [3], faithful and factually consistent generation [4], and reliable evaluation of semantic quality [5]. Despite progress in these areas, notable gaps remain, motivating the framework proposed in this dissertation.

1.2.1 Domain Adaptation and Representation Learning

Approaches to domain adaptation primarily rely on either pre-training models from scratch on domain-specific corpora or continuing the pre-training of general-purpose models [6, 7]. While these methods improve domain performance, they share a key limitation: they apply standard training objectives, such as Masked Language Modeling (MLM), which treat all tokens uniformly. This uniform weighting can be inefficient for specialized domains, where a small fraction of tokens representing domain-specific concepts carry disproportionate semantic weight.

Research has explored adaptive training schemes that modify which tokens are masked during training [8, 9], implementing intelligent masking strategies based on token frequency or domain relevance. However, these input-side modifications leave the loss computation unchanged: once a token is masked, prediction errors contribute equally regardless of the token's semantic importance. This limitation is particularly pronounced in specialized domains, where models must not only predict domain terminology but also learn the rich semantic relationships that structure expert knowledge.

Moreover, existing domain adaptation methods provide limited insight into the *nature* of the learned representations. Key open questions include whether domain-adapted models learn specialized, lower-dimensional representations or merely memorize surface-level patterns, and how the geometry and semantic organization of the embedding space evolve during domain-specific training. These questions remain critical for assessing the extent to which models capture domain semantics rather than merely exhibiting improved task performance through superficial adjustments.

1.2.2 Semantic Generation and Reasoning

Generating semantically faithful and comprehensive content in specialized domains poses distinct challenges beyond general text generation. For example, in question–answer generation from technical documents, systems must not only extract explicit information but also identify implicit relationships, synthesize multi-faceted concepts, and generate questions that probe different levels of understanding. Standard generation approaches, including in-context learning with Large Language Models (LLM) [1], may lack sufficient mechanisms for deep semantic reasoning, particularly in information-dense, technically complex contexts.

LLM-based multi-agent systems have demonstrated improved performance on complex reasoning tasks through collaboration, debate, and specialized role assignment [10, 11]. However, existing multi-agent frameworks primarily focus on task decomposition and voting mechanisms, insufficiently modeling the cognitive processes that enable deep understanding, preserve diversity, and ensure balanced convergence. Key open questions include: (i) how agents can engage in iterative reasoning processes that build upon each other's insights to uncover implicit relationships, (ii) how systems can balance consensus-seeking and diversity-preservation to ensure comprehensive coverage, and (iii) how collective intelligence principles from cognitive science and sociology can be operationalized to enhance semantic reasoning in NLP systems.

1.2.3 Evaluation Without Human Gold Standards

A central gap in specialized domain NLP is the evaluation bottleneck. High-quality human annotation requires domain expertise, making it both expensive and slow to obtain. This scarcity creates a self-reinforcing challenge: without adequate evaluation data, it is difficult to measure progress, compare systems, or identify failure modes.

Although multiple evaluation methods exist, their limitations suggest the need for approaches that more reliably assess semantic fidelity. *Self-consistency* approaches [12] can detect certain reasoning errors but do not directly evaluate whether outputs reflect source material. *Reference-free metrics* based on perplexity or other intrinsic measures tend to correlate weakly with semantic quality [13], and despite ongoing research efforts to improve these approaches [14], they remain inherently limited as primary evaluation methods. *LLM-as-judge* approaches [15] show promise but often rely on single-model judgments, raising concerns about reliability and potential bias. Collectively, these limitations highlight the need for evaluation methodologies that are both scalable and sensitive to the semantic relationships central to specialized domains.

A robust evaluation framework for specialized domains should satisfy several requirements: it should (i) be able to operate without extensive human annotation, (ii) leverage multiple models to reduce single-model bias, (iii) provide statistically principled reliability estimates, (iv) accommodate semantic equivalence despite lexical variation, and (v) generalize across domains and embedding representations. Few existing frameworks simultaneously meet all of these requirements, underscoring the necessity of new approaches.

1.3 Research Objectives and Questions

Guided by these challenges, this dissertation pursues three objectives:

1. **Representation**: Learn domain-aware representations by reweighting learning signals to emphasize domain-specific semantics and terminology during pre-training (Chapter 3).

2. **Reasoning**: Enable faithful and comprehensive generation on information-dense documents via collective, multi-agent workflows that balance diversity and convergence (Chapter 4).

3. **Evaluation**: Provide rigorous, interpretable, and reliable semantic evaluation without human gold standards through cross-model consensus and reliability analysis (Chapter 5).

These objectives give rise to the following research questions (RQs):

1. **RQ1 (Chapter 3)**: Can loss-side weighting for masked language modeling efficiently and robustly adapt language models to prioritize domain-specific semantics and terminology beyond uniform-loss or data-only adaptation approaches?

2. **RQ2 (Chapter 4)**: Which principles and mechanisms enable automated generation of question–answer pairs that are both comprehensive and faithful to technical context, particularly in settings requiring deep engagement and multifaceted reasoning?

3. **RQ3 (Chapter 5)**: How can we perform rigorous, interpretable, and reliable semantic evaluation of NLP systems in the absence of human-provided gold standards, leveraging semantic consensus and reliability estimation?

1.4 Research Contributions

This dissertation makes three primary contributions, each addressing a distinct challenge while forming part of an integrated methodology for advancing semantic fidelity in specialized domains.

1.4.1 Contribution 1: Adaptive Masked Language Modeling (Chapter 3)

We introduce Adaptive Masked Language Modeling (AMLM), a domain adaptation framework that shifts the focus of learning from input-side masking to output-side gradient modulation by applying importance weighting directly to the MLM loss. AMLM incorporates multiple contextual weighting strategies to prioritize tokens carrying significant semantic weight and employs stabilization mechanisms to ensure robust training under skewed weight distributions. This approach guides models toward specialized, semantically coherent, and compact representations, as demonstrated on financial domain tasks.

1.4.2 Contribution 2: Collective Intentional Reading (Chapter 4)

We introduce Collective Intentional Reading through Reflection and Refinement (CIR3), a multi-agent framework for generating comprehensive and faithful question–answer pairs. CIR3 formalizes comprehensiveness and faithfulness as a diversity-alignment objective and orchestrates specialized agents through iterative refinement, balancing coverage and grounding while reducing duplication and over-specificity in complex, information-dense documents.

1.4.3 Contribution 3: Consensus-Based Evaluation (Chapter 5)

We develop a consensus-based evaluation methodology for assessing semantic systems when human-annotated gold standards are unavailable. The framework establishes a stable semantic baseline through aggregation of outputs from multiple diverse language models and evaluates target systems based on their alignment with this induced consensus. Integrating multi-granularity clustering and reliability quantification, this approach provides a principled and practical alternative to traditional gold-standard evaluation in specialized domains.

1.4.4 Publications

Parts of this dissertation (Chapters 4 and 5) are based on the following peer-reviewed journal publication:

S. Saadaoui and E. Alonso, **"Coordinated LLM multi-agent systems for collaborative question-answer generation"**, *Knowledge-Based Systems*, vol. 330, p. 114627, 2025.

1.5 Scope

This dissertation focuses on textual modalities in two specialized domains: finance and medicine. While methods are designed to be domain-agnostic, empirical validation is limited to these domains. Finance and medicine were selected due to their high semantic density, economic and social importance, and availability of suitable corpora. Generalization to other specialized domains (e.g., law, scientific research) remains an important direction for future work. Limitations and potential threats to validity are discussed in Chapter 6.

1.6 Organization of the Dissertation

The remainder of this dissertation is organized as follows:

Chapter 2: Foundations and Literature Review surveys foundational concepts and prior research relevant to domain-adaptive representation learning, multi-agent reasoning for generation, and consensus-based evaluation, situating the contributions within the broader literature.

Chapter 3: Adaptive Learning for Domain Specialization introduces AMLM, presenting the weighted loss formulation, contextual weighting strategies, and experimental validation on financial domain tasks.

Chapter 4: Collective Reasoning for Faithful and Comprehensive QAG presents CIR3, formalizing the multi-agent coordination mechanisms and iterative refinement processes, with experimental evaluation in the financial and medical domains.

Chapter 5: Consensus Without Gold presents the consensus-based evaluation methodology, detailing procedures for establishing semantic consensus and assessing target systems, with cross-domain validation and analysis of robustness to different embedding representations. The chapter also demonstrates how this framework provides an independent validation layer for the other contributions.

Chapter 6: Conclusion and Future Work synthesizes the dissertation's findings, discusses scope and limitations, and outlines directions for future research in semantic fidelity for specialized domains.

Chapters 3–5 are relatively self-contained, with methodology, experiments, and results. Taken together, the three contributions provide an end-to-end perspective for semantic fidelity in specialized domains: AMLM delivers domain-adapted representations, CIR3 enables faithful and comprehensive generation, and the consensus approach evaluates representation quality and generation fidelity without gold standards. Chapter 5 cross-validates AMLM and assesses CIR3, demonstrating interoperability while preserving independence of each contribution.

Chapter 2

Foundations and Literature Review

This chapter synthesizes the literature underpinning the core contributions of this dissertation. Section 2.1 reviews foundational concepts. Section 2.2 examines representation learning for domain specialization (Chapter 3); Section 2.3 examines multi-agent reasoning for comprehensive and faithful QA generation (Chapter 4); and Section 2.4 examines consensus-based semantic evaluation without gold standards (Chapter 5).

2.1 Foundational Concepts

The transformer architecture [16] revolutionized NLP by replacing recurrent mechanisms [17, 18, 19] with self-attention. This design enables parallel processing and captures long-range dependencies. Modern language models adopt variants of this architecture depending on their intended use: encoder-only models (e.g., BERT [2]) excel at understanding tasks like classification and named entity recognition; decoder-only models (e.g., GPT [20, 1]) generate text autoregressively; encoder-decoder models (e.g., T5 [21], BART [22]) combine both for sequence-to-sequence tasks such as translation, summarization, question answering, and conditional text generation.

2.1.1 Pre-training Objectives

Self-supervised pre-training enables models to learn rich language representations from large-scale unlabeled text. Common objectives include:

- *Masked Language Modeling (MLM)* [2], which randomly masks tokens in input sequences and trains the model to predict them, enabling bidirectional context modeling.
- *Next Sentence Prediction (NSP)* [2] and its alternative *Sentence Order Prediction (SOP)* [23], which target inter-sentence coherence. However, empirical results suggest that NSP provides limited downstream gains [24].
- *Span-based objectives* [25], which require the model to reconstruct contiguous masked spans, encouraging it to capture longer-range contextual dependencies.
- *Causal language modeling*, predicting tokens sequentially based on previous tokens, used in autoregressive models like GPT [26].
- *Denoising objectives* (T5 [21], BART [22]), which corrupt contiguous spans or whole segments of input text (by token masking, deletion, or permutation) and train models to reconstruct the original, blending masked token prediction with sequence generation.
- *Contrastive learning* [27, 28, 29], which brings semantically similar instances closer in embedding space while separating dissimilar ones, improving unsupervised and semi-supervised representation quality. Recent approaches adapt contrastive objectives for both encoder-only and encoder-decoder architectures in LLMs [30].

AMLM builds on masked language modeling, with principles that extend to other pre-training objectives.

2.1.2 Foundation Models

The rapid evolution of foundation models has produced increasingly capable and specialized systems. Model families such as Llama 3/3.1 [31], Qwen 2/2.5/3 [32, 33, 34, 35], and Phi-3 [36, 37] illustrate advances in data curation, scaling, and instruction tuning. Open-weight variants (e.g., Llama 3.1, Qwen 2.5, Gemma 3 [38]) serve as standard baselines for benchmarking and downstream adaptation, while closed-weight systems such as Claude 4 and GPT-4/4o [39, 40] represent state-of-the-art capabilities, integrating multimodal reasoning, extended context handling, and refined safety alignment.

Iterations including GPT-4o, Claude 4, and Gemini 2.5 [41] increasingly unify text, vision, and audio modalities within single generative frameworks. These models also underpin benchmark ecosystems such as HELM 2.0 [42, 43], LMSYS Arena [44, 45], and EvalPlus [46, 47], providing consistent reference points for evaluation.

Parallel work explores architectural innovations, including Mixture-of-Experts (MoE) architectures [48, 49, 50], which dynamically route computation across specialized subnetworks to improve efficiency. Examples include DeepSeek-V2 [51] and Gemini 2.5. The divergence between open- and closed-weight releases also informs discussions of transparency, reproducibility, and model governance, reflecting tensions between openness and proprietary control in foundation model development.

This work employs different categories of foundation models to match the requirements of each pillar. Encoder-only models (BERT variants) support domain-specific representation learning (Chapter 3); decoder-only, instruction-tuned models (e.g., Llama 3, Gemma 3) enable multi-agent QAG, leveraging autoregressive reasoning for controlled outputs (Chapter 4); and multi-model ensembles, comprising both open-weight and closed-weight systems, facilitate consensus-based evaluation (Chapter 5).

2.1.3 Evaluation Paradigms

Pre-trained representations can be evaluated either intrinsically or extrinsically. *Intrinsic evaluation* assesses properties of the embeddings directly, for example through perplexity on held-out corpora, probing models for linguistic phenomena [52], or geometric analysis of embedding spaces. *Extrinsic evaluation* measures practical performance on downstream tasks (e.g., question answering, sentiment analysis, natural language inference) through fine-tuning or feature-based transfer. In fine-tuning, model parameters are updated to optimize task-specific objectives, whereas feature-based transfer keeps the pre-trained model fixed and trains only a lightweight classifier on top of the frozen embeddings [53]. While extrinsic metrics reflect task-specific utility, intrinsic measures provide insight into the knowledge encoded within representations. Section 2.4 and Chapter 5 extend these paradigms to multi-model consensus and reliability estimation.

2.2 Representation: Domain-Specific Encoders

Domain adaptation addresses the challenge of transferring pre-trained models to contexts where vocabulary, style, and conceptual relationships differ from general-purpose corpora. Strategy choice depends on corpus size, domain shift, and computational resources. This section reviews primary adaptation techniques, highlighting gaps that motivate the AMLM approach (Chapter 3). The discussion proceeds from pre-training choices, through corpus, task, and vocabulary refinements, to adaptive masking, and parameter-efficient adaptation.

2.2.1 Pre-training: From Scratch vs. Continued Pre-training

Domain adaptation is typically achieved either by pre-training a model from scratch on a domain-specific corpus or by continuing the pre-training of a foundational model on the target domain data [54, 55]. Pre-training from scratch allows all layers, including tokenization and embeddings, to specialize immediately on domain-specific patterns, yielding strong in-domain performance if sufficient data and compute are available [56]. Continued pre-training leverages linguistic generalization from a foundational model and adapts the model's weights to the target corpus. This approach is typically more resource-efficient [3], and because lower layers tend to remain stable while higher layers adapt to the new domain, it can reduce the likelihood of catastrophic forgetting[1] [59].

BioBERT [6] exemplifies continued pre-training in the biomedical domain, achieving state-of-the-art results in named entity recognition, relation extraction, and question answering. Similarly, FinBERT [7] adapts BERT for financial text, improving sentiment analysis and classification tasks. These examples demonstrate that large-scale domain adaptation can markedly boost downstream performance.

Despite their effectiveness, conventional pre-training strategies do not explicitly prioritize in-domain tokens during loss computation, motivating approaches that emphasize domain-specific terminology.

[1]Catastrophic forgetting [57, 58] occurs when training on new tasks overwrites previously learned information. It is a central challenge for continual learning and domain adaptation, especially when domain data is limited.

2.2.2 Corpus, Vocabulary, and Task Adaptation

Domain adaptation can be refined along three axes: corpus scope, task locality, and tokenizer coverage.

Domain-Adaptive Pre-training (DAPT) [3] is the standard instantiation of continued pre-training on unlabeled in-domain text and generally yields consistent gains under substantial distribution shift. This approach broadens the model's exposure to the general characteristics of a domain, making it more robust to shifts in vocabulary, style, and topic. **Task-Adaptive Pre-training (TAPT)** [3] narrows adaptation to unlabeled data from a specific downstream task, complementing DAPT when task data diverges stylistically from the domain. TAPT refines the model by focusing on task-specific patterns present in the task's unlabeled data, which can be especially beneficial when the task's data distribution is not fully captured by the broader domain corpus. Domain-specialized models such as BioBERT [6], SciBERT [60], FinBERT [7], and LegalBERT [56] exemplify the practical impact of these adaptation strategies, each tailored to the linguistic and structural demands of their respective domains.

The choice of tokenizer and its coverage also plays a critical role, as it determines how well the model can represent and generalize across different types of text, particularly in specialized or low-resource settings. **Vocabulary adaptation** reduces subword fragmentation for technical terminology. Domain tokenizers, or selective vocabulary expansion methods like VE-GAD [61], improve coverage while minimizing parameter growth (primarily new embedding vectors). Practical trade-offs involve corpus selection, computational cost, and strategies to limit catastrophic forgetting, such as mixing general-domain text or applying continual learning regularizers [57, 58].

Orthogonal to these input-side choices, alternative approaches can emphasize domain-critical tokens during training while complementing DAPT/TAPT strategies.

2.2.3 Adaptive Approaches in Masked Language Modeling

While the approaches discussed above primarily target static domain adaptation via continual pre-training on domain-specific corpora, dynamic adaptation strategies enhance masked language modeling by prioritizing more salient or informative tokens. Adaptive masking policies dynamically adjust token masking probabilities based on criteria such as token difficulty, frequency, domain relevance, or position within a sequence [62]. Techniques like Weighted Sampling

(WSBERT) [8] and scheduled masking [9] modulate masking distributions to mitigate frequency bias and improve model generalization. Representative strategies include dynamic masking, as implemented in RoBERTa [24], which regenerates mask patterns each epoch to increase data diversity, exposing the model to a wider variety of masked inputs and enhancing its ability to learn robust context representations. Another example is knowledge- and entity-aware masking, such as ERNIE's approach [63], which prioritizes masking of phrases and named entities to better capture domain-specific semantics and relationships.

Beyond modifying which tokens are masked, an alternative approach is to adjust how prediction errors are weighted during training, allowing selective emphasis on domain-specific terminology.

2.2.4 Parameter-Efficient Adaptation

Parameter-Efficient Fine-Tuning (PEFT) methods such as Adapters [64] and Low-Rank Adaptation (LoRA) [65] adapt LLMs by training only small additional modules or low-rank matrices while keeping most of the original parameters frozen. This approach significantly reduces both computational requirements and memory usage during fine-tuning. Despite training far fewer parameters, these methods typically achieve strong downstream task performance comparable to full fine-tuning.

Parameter-efficient methods are orthogonal and complementary to training objective modifications. Combining architectural efficiency with adaptive training objectives represents a promising direction for domain specialization.

2.2.5 Guidance, Stability, and Regularization

Beyond corpus and masking choices, several strategies guide model focus and maintain stability during domain specialization.

Fine-Tuning Strategies. Gradual unfreezing and discriminative learning rates [66] improve stability during domain adaptation. These methods provide a foundation for safer task-specific adaptation and complement curriculum and weighting techniques.

Curriculum and Importance Weighting. Curriculum learning organizes training data from easy to hard examples, dynamically adjusting sample importance over time to guide learning progression [67]. In contrast, focal loss statically down-weights easy samples throughout training, concentrating on harder, misclassified instances to address class imbalance [68]. Both methods enhance learning by emphasizing challenging data but differ fundamentally in their dynamic versus fixed importance weighting schemes.

Stability and Regularization. A persistent challenge in continued pre-training is *catastrophic forgetting*: the loss of general-language capability when a model is adapted to a narrow domain.

- **Regularization-based methods** restrict parameter drift to preserve pretrained knowledge and enhance training stability. Elastic Weight Consolidation (EWC) [58] penalizes updates to critical parameters for prior tasks, while Mixout [69] stochastically replaces fine-tuned parameters with pre-trained ones to improve stability. Label smoothing [70] curbs overconfidence in predictions during training phase. Dropout [71] randomly deactivates a subset of neurons during training, which forces the network to not rely heavily on any individual neuron. This encourages the network to develop redundant representations across neurons, improving robustness and generalization, while mitigating catastrophic forgetting in continual learning.
- **Residual connections** [72] introduce shortcut paths that enable layers to learn modifications to the input rather than entirely new transformations, facilitating gradient flow, stabilizing training, and enabling very deep architectures. By supporting identity mappings, residual connections allow the network to retain prior representations while learning new domain-specific refinements, thus improving robustness and mitigating catastrophic forgetting. Residual connections are a fundamental component in transformer-based models and are widely adopted across domain adaptation and continual learning settings.
- **Normalization methods**, including Layer Normalization [73] and Batch Layer Normalization [74], maintain consistent activation distributions and smooth gradients, reducing disruptive parameter shifts.
- **Replay-based methods** interleave domain-specific data with general-domain examples to preserve broad linguistic knowledge [75]. Gradient Episodic Memory (GEM) methods constrain gradient updates to prevent interference with previously learned tasks [76].
- **Knowledge distillation** [77] transfers the softened outputs of a general-domain teacher model to a domain-adapted student, preserving generalization while adapting to new domain patterns.

- **Parameter-efficient fine-tuning** updates only small subsets of parameters, such as adapter modules or low-rank factorization methods (LoRA), which preserves pretrained weights and minimizes forgetting [64, 65].
- **Dynamic architecture methods**, such as Progressive Neural Networks (PNN), modify or expand model structure progressively to incorporate new domain knowledge without overwriting existing capabilities [78].
- **Adversarial and robust training** injects perturbations during training to improve model generalization and stability across domains [79].
- **Ensemble and mixture of experts** approaches combine multiple specialized models or modules to retain general language understanding while adapting to diverse domains [48].

Together, these complementary strategies enable continued pre-training methods to balance adaptation and retention, ensuring improved domain specialization without compromising a model's general understanding.

While existing methods improve adaptation, they typically rely on static heuristics (fixed curricula, constant weighting, predefined masking) that treat all tokens uniformly during training. Chapter 3 addresses this limitation by introducing adaptive masked language modeling, which dynamically modulates the learning signal to emphasize domain-relevant terminology based on contextual and distributional importance, while stabilization mechanisms ensure robust training under skewed weight distributions.

2.3 Reasoning: From ICL to Single- and Multi-Agent Control

Shifting from representation learning to generation, this section examines reasoning systems, progressing from single-agent methods to multi-agent coordination. While domain-adapted encoders capture specialized knowledge (Section 2.2), generating comprehensive and faithful content from that knowledge requires coordinated reasoning. Single-model approaches, including chain-of-thought prompting, often struggle with complex, information-dense documents: they may overlook key concepts (insufficient comprehensiveness) or hallucinate details not grounded in the source text (reduced faithfulness). This limitation is particularly acute in Question–Answer Generation (QAG), where the objective is to produce question–answer pairs that fully and faithfully reflect a given context. Multi-agent systems provide a paradigm to address these limitations by leveraging collective intelligence. By structuring collaboration among multiple LLM agents, such systems can simulate diverse viewpoints, facilitate critical debate, and iteratively refine outputs toward deeper understanding of the source material.

This section reviews the QAG task and its evolution (Section 2.3.1), In-Context Learning as a foundational conditioning mechanism (Section 2.3.2), single-agent reasoning methods (Section 2.3.3), and multi-agent collaboration patterns with control and termination mechanisms (Sections 2.3.5, 2.3.6).

2.3.1 Question–Answer Generation (QAG)

Question–Answer Generation (QAG) is the task of generating question–answer pairs from a given context document. It underpins a wide range of applications, from enhancing information retrieval systems [80, 81] and building educational tools [82, 83] to augmenting datasets for downstream QA training [84].

QAG is related to, but distinct from, its sub-tasks: Question Generation (QG), Answer Extraction (AE), and Question Answering (QA):

- **Question Generation (QG)** focuses on generating a question given a context and a pre-selected answer span, explored through both rule-based [85, 86] and neural [87, 88] methods.
- **Answer Extraction (AE)** or Machine Reading Comprehension (MRC) identifies an answer span from a text given a question [89, 90].
- **Question Answering (QA)** is the general task of responding to a question, optionally conditioned on a provided document.

In contrast, QAG is a holistic and more challenging task, as it must identify salient information worth questioning and then generate both the question and a faithful answer. This requires deeper text understanding than QG or AE alone.

Evolution of QAG Models. Early approaches were rule-based or statistical, relying on syntactic parsing and handcrafted templates to transform declarative sentences into questions [85, 86]. Neural sequence-to-sequence models subsequently enabled data-driven question generation [87], often trained on reading comprehension datasets such as SQuAD [89].

The rise of pre-trained language models (PLMs) marked a decisive shift: models like BERT [2] were fine-tuned for AE and QG components in pipeline systems [91], while encoder-decoder models such as BART [22] and T5 [21] enabled end-to-end QAG by jointly generating question and answer pairs conditioned on the context [84, 92].

Limitations and Open Challenges. Despite progress, QAG faces persistent challenges, particularly for complex, domain-specific documents:

- **Lack of Comprehensiveness:** Generated QA pairs often capture only surface-level facts, neglecting implicit relationships and higher-order concepts [93]. This limitation is attributed partly to the absence of multi-reference training datasets that exhaustively cover all possible questions for a context [94].
- **Lack of Faithfulness:** Generative models, especially large-scale ones, are prone to hallucination [10, 95], producing answers that are plausible but not factually grounded in the source text. This is particularly problematic in high-stakes domains like finance or medicine.
- **Limited Question Diversity:** Models tend to generate repetitive or semantically similar questions [96], overemphasizing simple *Wh*-types at the expense of complex reasoning questions.

Research has attempted to address these issues. For example, [97] and [96] enriched generation with entity-level metadata to improve coverage, while [94] proposed dynamic identification of question-worthy content before generation. Yet balancing comprehensiveness with faithfulness remains difficult, especially in specialized domains. These limitations motivate structured reasoning frameworks such as the multi-agent systems discussed next. Chapter 4 addresses the challenge of balancing comprehensiveness with faithfulness.

2.3.2 In-Context Learning (ICL): Foundations and Implications

In-Context Learning (ICL) is an emerging capability[2] in which models adapt their behavior through natural-language instructions and a small set of demonstrations within the prompt,

[2] An *emergent capability* refers to a novel or unexpected behavior exhibited by large-scale models that was not explicitly trained for. Such abilities often arise non-linearly, emerging only once a critical scaling threshold in model size, training data, or compute is reached, as observed in scaling law studies [98]. Examples include complex multi-step reasoning (e.g., Chain-of-Thought prompting) and tool use or code generation.

without any parameter updates [1]. ICL is especially valuable in low-resource or rapidly evolving domains, as it encodes task structure at inference time. However, its performance is highly sensitive to demonstration selection, order, and format, and it can exhibit calibration biases that distort prediction distributions [99].

For information-dense tasks such as QAG, plain ICL often under-explores implicit relations, motivating additional scaffolds such as elicited reasoning, external tools, or multi-agent collaboration. ICL serves as the foundational conditioning mechanism in this chapter, while subsequent sections examine superstructures that enhance comprehensiveness (coverage of salient concepts) and faithfulness (grounding in the source). Chapter 4 builds on ICL through structured multi-agent coordination.

2.3.3 Single-Agent Reasoning: Elicitation, Planning, Search, and Tools

While LLMs exhibit improved reasoning when prompted to articulate intermediate steps or aggregate multiple reasoning paths, structuring them as *agents* enables autonomous, goal-directed decision-making through interaction with their environment, other agents, or external tools [100]. Single-agent approaches can then be categorized along several axes: (i) reasoning elicitation, (ii) explicit planning, (iii) search over intermediate states, (iv) tool use and externalized computation, and (v) self-reflection and verification.

Chain-of-Thought (CoT) Prompting. Prompting models to generate explicit reasoning steps enhances performance on arithmetic, commonsense, and symbolic reasoning tasks [101]. CoT converts implicit reasoning into observable token sequences, enabling models to perform multi-hop inference that would fail under direct question–answer prompting. Zero-shot CoT [102] further demonstrated that such reasoning behaviors can emerge from simple natural-language triggers, such as “Let’s think step by step”.

Self-Consistency. [12] improved CoT reasoning by sampling multiple reasoning paths at a high temperature and selecting the most frequent answer through majority voting. This approach demonstrates that marginalizing over diverse reasoning trajectories, rather than relying on greedy decoding or single-sample CoT prompting, leads to more reliable and consistent results on well-defined tasks. The method assumes that correct answers emerge more consistently across diverse reasoning paths than incorrect ones, an assumption that holds for well-defined problems but may break down for subjective or ambiguous tasks.

Planning and Search. Plan-and-Solve prompting [103] separates planning from execution, enhancing the stability of zero-shot reasoning. Tree of Thoughts (ToT) [104] generalizes this idea into a structured search over intermediate reasoning states, enabling backtracking and lookahead. Graph-based extensions such as Graph of Thoughts [105] further support flexible composition and integration of partial reasoning states. Self-Discover [106] allows models to autonomously design their own reasoning scaffolds prior to problem solving, thereby reducing the need for manual prompt engineering.

Tool Use and Externalization. Several approaches leverage tools or external computation to enhance reasoning and efficiency. ReAct [107] interleaves reasoning steps with tool actions, while ReWOO [108] decouples planning from observation gathering, reducing token usage and improving robustness. Program- or Code-of-Thoughts methods externalize computation by generating executable code or structured programs [109]. Related precedents include Program-Aided Language Models (PAL) [110] and Toolformer [111], which learn tool policies and calculator/function calls to offload symbolic work.

Parallel execution and structured search further improve efficiency and reasoning quality. Parallel function-calling compilers reduce latency and orchestration overhead by planning and executing tool calls concurrently [112]. Language Agent Tree Search (LATS) [113] unifies planning, acting, and search over trajectories within a single agent, bridging single-agent reasoning with tree-based exploration.

2.3.4 Retrieval-Augmented Generation (RAG) and Grounding

A complementary strategy for improving faithfulness and mitigating hallucination involves augmenting generation with retrieved evidence from external knowledge sources. Retrieval-Augmented Generation (RAG), introduced for knowledge-intensive NLP tasks [114], conditions generative models on passages retrieved from a large corpus. This paradigm grounds outputs in verifiable sources and enables attribution, addressing key limitations in purely parametric generation. Dense retrieval forms the foundation of modern RAG systems. Dense Passage Retrieval (DPR) [115] learns dual encoders for queries and documents through contrastive objectives, which supports semantic retrieval via maximum inner product search. REALM [116] jointly trains retrieval and language modeling components by treating document selection as a latent variable. Late-interaction models such as ColBERT [117] and ColBERTv2 [118] further balance retrieval quality and computational efficiency by computing token-level interactions only after initial candidate pruning. Recent work has focused on improving the reliability and controllability of retrieval-conditioned generation. Self-RAG [95] trains models to generate special reflection tokens that assess retrieval necessity and output quality, which enables adaptive retrieval and self-correction. Corrective RAG (CRAG) [119] evaluates the relevance of retrieved evidence and applies corrective strategies, including web search augmentation, for low-confidence retrievals. RAPTOR [120] constructs hierarchical summaries over retrieved passages to capture multi-scale semantic structure. SimRAG [121] proposes a self-training approach for domain adaptation, where LLMs generate synthetic, domain-specific queries and high-quality QA pairs are used to jointly fine-tune retrieval and generation. Survey work [80, 122] synthesizes these developments across retrieval architectures, indexing strategies, and retrieval-generation interaction patterns.

RAG can enhance both comprehensiveness and faithfulness. Retrieved passages expand coverage of relevant concepts, while grounding in source documents mitigates hallucination. However, RAG introduces dependencies on retrieval quality, and irrelevant or misleading evidence can adversely affect generation. Context window constraints and the integration of retrieved evidence with structured multi-agent reasoning also remain open research challenges [80].

Complementary to retrieval-augmented approaches, when documents are self-contained, Chapter 4 addresses this scenario by operating under a closed-context assumption, where all relevant information is contained within the input context.

Self-Reflection and Verifier-Augmented Reasoning. Beyond external plans and tools, LLMs can improve reasoning through internal evaluation, critique, and feedback loops. Reflexion [123] employs verbal reinforcement-style feedback to revise solutions, while Self-Refine [124] iteratively critiques and improves outputs. However, these mechanisms can degrade performance if applied indiscriminately when initial answers are already correct, underscoring the need for calibrated control.

Verifier-augmented methods extend this idea by pairing generators with verifiers or process-reward models (PRMs) that evaluate intermediate reasoning steps or final answers. By scoring each step or ranking alternative reasoning chains, these approaches enable step-level validation, error correction, and guided refinement, improving the reliability and accuracy of generated reasoning paths [125, 126, 127, 128]. Beyond prompting-based methods, specialized instruction-tuned or verifier-augmented models explicitly train for stepwise deliberation and verification.

While these methods enhance individual reasoning robustness, they remain constrained by single-agent capacity and lack explicit cross-perspective verification, gaps that are addressed by multi-agent reasoning.

2.3.5 Multi-Agent Reasoning: Collaboration Patterns and Frameworks

Multi-agent reasoning leverages epistemic diversity and complementary expertise through role specialization and structured interaction protocols, enabling more robust and thorough reasoning. Debate frameworks [10] encourage agents to critique one another, while role differentiation enhances exploration of alternative reasoning paths [11].

General frameworks include conversational orchestration (AutoGen [129]), role-play-based cooperation (CAMEL [130]), and procedure-driven pipelines (MetaGPT [131]). Search-centric orchestrators such as LATS [113] bridge single- and multi-agent paradigms by exploring action and plan trees with agent-in-the-loop evaluation.

Key design dimensions include topology (centralized vs. decentralized communication), memory (shared buffers vs. selective retrieval), convergence mechanisms (voting, moderation, verifiers), and diversity maintenance (role heterogeneity, sampling temperature).

Multi-Agent Debate and Role Differentiation. Multi-agent debate [10] involves multiple LLM instances generating answers, critiquing each other, and iteratively refining their positions until convergence or a maximum number of rounds. This process leverages reasoning diversity, as agents explore different reasoning paths and critique weak arguments, improving accuracy over self-consistency, particularly when initial answers are incorrect. Role specialization [11] further increases reasoning diversity by assigning agents distinct perspectives (e.g., "skeptic" vs. "advocate"), reducing premature convergence. However, this approach requires careful prompt engineering to maintain roles across iterations.

Diversity vs. Convergence Trade-off. Multi-agent systems face a tension between exploration and exploitation. Diversity mechanisms, such as high-temperature sampling or independent agents, enable exploration of alternative solutions and guard against overconfidence but may fail to reach consensus. Convergence mechanisms, including majority voting [132] or greedy decoding, produce decisive answers but risk amplifying initial biases. Effective systems dynamically balance cognitive diversity with factual alignment to optimize reasoning performance.

Consensus and Coordination Mechanisms. Beyond debate and critique, several works explore explicit consensus-seeking and coordination protocols. Inter-consistency through inter-agent negotiation [133] iteratively adjusts positions to reduce disagreement. Inspired by multi-robot collaboration, [134] examined how agent number, personality, and network topology affect consensus in a constrained 1D space. [135] explored flocking behaviors, where agents maintain proximity while avoiding collisions and preserving formations, demonstrating how local coordination rules can produce emergent collective behavior. These studies highlight the importance of explicit convergence mechanisms and topological considerations in multi-agent reasoning systems. However, balancing diversity preservation with convergence in high-dimensional semantic spaces remains an open challenge.

2.3.6 Control, Memory, and Termination

Multi-agent systems require mechanisms for managing shared context, coordinating iterative reasoning, and determining when to terminate refinement. This subsection reviews control structures and memory architectures that support collaborative reasoning.

Transactive Memory Systems (TMS). [136] describe how groups collectively encode, store, and retrieve knowledge. In human teams, individuals often specialize in different knowledge domains and maintain meta-knowledge of who knows what, enabling efficient distributed cognition. [137] formalized three key TMS dimensions: *specialization* (differentiation of expertise among members), *credibility* (trust in the expertise of others), and *coordination* (the team's ability to integrate and apply distributed knowledge effectively).

These principles provide an analogy for multi-agent LLM architectures. Agents can specialize in different reasoning strategies, maintain meta-knowledge of which agent produced which output, and coordinate their access to and integration of shared context [138, 139, 140].

Shared Context Management. Multi-agent LLM systems employ several strategies for managing shared information:

- **Message-passing architectures**: Agents communicate via explicit message exchanges, maintaining individual contexts while selectively incorporating outputs from others. Multi Agent Debate (MAD) implemented this in debate frameworks, where each agent observes the previous round's responses from all participants [10].

- **Centralized context**: All agents share access to a global memory buffer that stores every intermediate output throughout the workflow. This design ensures complete information availability at each step, facilitating multi-agent coordination and complex reasoning. However, the accumulation of outputs over many iterations can risk exceeding the language model's context window or memory limits [129, 131, 113].

- **Selective attention**: Agents dynamically retrieve only the most relevant prior outputs, leveraging semantic similarity search, embedding-based retrieval, or structured queries. This selective approach improves scalability and efficiency, but depends on robust retrieval and relevance-ranking mechanisms to ensure accurate context selection for each agent's task [107, 108, 141, 142].

Stopping Criteria and Convergence Detection. Iterative reasoning systems require principled termination conditions to ensure efficiency and reliability: stopping too early may leave reasoning incomplete, whereas continuing unnecessarily wastes computational resources. Common approaches include:

- **Fixed iteration budgets**: Run a predetermined number of refinement rounds, as used in [124, 10]. This method offers predictable computational cost and requires no convergence monitoring, but may stop prematurely or continue unnecessarily.

- **Answer stability**: Monitor whether outputs change between iterations. If consecutive rounds produce identical or highly similar answers, convergence is assumed. For open-ended or multi-answer tasks, similarity can be measured using embedding-based metrics or semantic matching [143].

- **Quality thresholds**: Continue iterating until outputs satisfy predefined quality criteria, such as confidence scores or validator agreement. This requires reliable estimators or validators, which may not always be available for open-ended tasks [127, 144, 145].

- **Composite criteria**: Combine multiple signals. For instance, stop when either the maximum number of iterations is reached, or answer stability is high and output meets quality thresholds or validator agreement. This approach provides robustness against failures of individual criteria [113, 10, 133].

Iteration Control Challenges. Determining the optimal iteration depth in iterative reasoning systems is non-trivial. Too few iterations underutilize the system's refinement capacity, while too many waste computation and risk performance degradation if agents introduce errors. Adaptive control (adjusting iteration budgets based on task difficulty or intermediate quality signals) remains an open problem. Additionally, as context windows fill with prior iterations, later rounds may lose access to original inputs, further degrading performance.

This section reviewed reasoning systems from ICL foundations through single-agent methods to multi-agent coordination frameworks. The literature evolves from elicited individual reasoning (CoT, self-consistency, planning) to structured multi-agent deliberation under explicit control and memory. Key challenges for comprehensive and faithful QAG include: (i) balancing perspectival diversity with convergence to semantic alignment; (ii) formalizing transactive memory and shared-context protocols for coordinated reasoning; and (iii) developing principled termination criteria that avoid both premature convergence and unnecessary computation. Chapter 4 addresses these challenges through structured multi-agent coordination with explicit convergence mechanisms and principled termination criteria.

2.4 Evaluation Without Gold: Consensus and Reliability

Having examined representation and generation, we turn to evaluation, the third pillar of semantic fidelity. Specialized domains often lack gold-standard annotations, which necessitates alternative evaluation paradigms. This section reviews LLM-based evaluation approaches and consensus mechanisms with reliability measures, and motivates the consensus framework developed in Chapter 5.

2.4.1 LLM-as-Judge and Limitations

Traditional NLP evaluation relies on human-annotated gold standards to measure system performance. However, obtaining high-quality annotations is costly and time-consuming, particularly in specialized domains where expert annotators are scarce. This challenge has motivated research into alternative evaluation paradigms.

Evaluation Paradigms Without Human Annotation. Several approaches attempt to assess quality without human labels. Reference-free metrics evaluate quality based on intrinsic properties of generated text. BLANC [146] measures summarization quality through language model performance changes when using the summary as context; better summaries improve downstream LM performance. MAUVE [147] quantifies distributional divergence in open-ended generation by comparing embedding distributions between generated and reference corpora. However, calibration to human judgment varies substantially across tasks and domains.

LLM-Based Evaluators. Research has explored using LLMs as evaluators across various NLP tasks:

- **GPT-4 as evaluator** [15]: Chatbot Arena used GPT-4 to judge pairwise comparisons, reporting agreement with human preferences alongside documented biases (position, verbosity, self-enhancement: preferring outputs from the same model family).
- **G-Eval** [148]: Rubric-then-score prompting improves correlation with human ratings on summarization and dialogue tasks.
- **Prometheus / Prometheus 2** [149, 144]: Open evaluators trained on feedback data, reporting improved robustness as open alternatives to proprietary judges.
- **AgentBench** [150]: Evaluates LLMs as agents across multiple dimensions (reasoning, planning, tool use), using model-generated assessments to rank capabilities. Results highlight that evaluation quality depends heavily on task specification and rubric clarity.
- **Open-weight evaluators**: Commonly used open-weight families include Llama 3.1, Qwen 2.5, and Gemma 3 27B (instruction-tuned) [151, 152, 153].

Limitations and Reliability Concerns. While LLM-as-judge approaches reduce annotation costs, they introduce new challenges:

- **Model-specific biases**: Single-model judgments reflect training data artifacts and systematic preferences. [15] documented presentation-order effects, where judgment quality depends on whether the superior response appears first or second. Follow-up analyses and judge benchmarks (e.g., RewardBench [145]) further quantify brittleness and bias in scorer models and LLM judges.
- **Prompt sensitivity**: Evaluation quality varies with prompt formulation, evaluation criteria specificity, and output format (binary, Likert scale, natural language). Small prompt changes can yield inconsistent judgments.

- **Lack of reliability mechanisms**: Single-model evaluation provides no mechanism for quantifying judgment reliability or consensus. Unlike human annotation studies, which report inter-rater agreement statistics (e.g., Kappa, Alpha), LLM evaluations typically lack reliability estimates.
- **Calibration challenges**: LLM confidence scores (e.g., output probabilities) correlate poorly with judgment accuracy. Models may express high confidence in incorrect evaluations.

These limitations motivate multi-model consensus approaches that aggregate judgments from multiple evaluators, enabling reliability assessment and reducing model-specific biases.

Consensus Aggregation Methods. Several approaches aggregate judgments across multiple models. ChatEval [132] explores voting and debate schemes among LLMs for discrete judgments, operating on categorical labels or exact matching. While these methods demonstrate the value of multi-model consensus, they do not integrate explicit reliability quantification, nor do they systematically validate agreement across diverse representations for open-ended generative tasks.

2.4.2 Agreement Measures and Semantic Aggregation

Evaluating generative systems requires mechanisms for aggregating multiple judgments and quantifying agreement reliability. We review inter-rater reliability metrics, semantic clustering for label induction, and matching strategies for comparing generated outputs.

Inter-Rater Reliability (IRR) Metrics. IRR metrics quantify the degree of agreement among multiple annotators, providing statistical measures that account for chance agreement. [154] provided a comprehensive survey distinguishing between observed agreement (percentage agreement, which ignores chance) and chance-corrected metrics:

- **Cohen's Kappa** [155]: Measures agreement between two raters on nominal categories, adjusting for expected chance agreement. Values range from -1 (perfect disagreement) to 1 (perfect agreement), with 0 indicating chance-level agreement. Kappa is widely used but limited to two raters.
- **Fleiss' Kappa** [156]: Extends Cohen's Kappa to multiple raters, computing agreement across all rater pairs. Requires complete annotations: all raters must judge all items.

This assumption limits applicability when annotators evaluate different subsets or when missing data exists.

- **Krippendorff's Alpha** [157]: A generalized agreement measure that handles missing data, arbitrary numbers of annotators, and different measurement scales (nominal, ordinal, interval, ratio). Alpha is particularly suited for consensus assessment in LLM evaluation, where different models may evaluate different subsets of outputs or fail to produce valid judgments for some inputs.

Prior work has applied Krippendorff's Alpha to LLM evaluation by computing agreement on discrete model-generated labels [158]. While Alpha accommodates multiple measurement levels (nominal, ordinal, interval, ratio), it is not defined for high-dimensional embeddings. This raises a challenge for generative evaluation: how can chance-corrected IRR be assessed when outputs lack predefined categories and semantically equivalent responses differ lexically? While Krippendorff's Alpha has been used for discrete labels, its application to high-dimensional embeddings in generative evaluation remains largely unexplored. Chapter 5 addresses this gap by extending reliability quantification to open-ended generative outputs through semantic clustering approaches.

Semantic Clustering and Label Induction. Several clustering techniques enable grouping generative outputs into semantic categories:

- **Density-based clustering**: DBSCAN [159] and HDBSCAN [160] identify clusters of varying density without requiring predefined cluster counts. These methods mark outliers as noise, useful for filtering low-quality or off-topic outputs. However, performance depends heavily on distance threshold tuning.

- **Hierarchical clustering**: Agglomerative clustering [161] builds a hierarchy of clusters by iteratively merging the closest pairs. Cutting the dendrogram at different heights yields multi-granularity clusterings. Distance thresholds control semantic specificity: low thresholds produce fine-grained topics, high thresholds yield broad categories.

- **Topic modeling**: [162] explored clustering for topic discovery in dialogue using sentence transformers with hierarchical clustering to identify latent thematic structure, demonstrating that embedding-based clustering captures semantic relationships missed by lexical methods.

Once outputs are clustered, cluster assignments serve as induced labels for IRR computation. This enables reliability assessment for generative tasks where gold labels do not exist.

Pairwise Semantic Matching. Complementing holistic clustering, pairwise semantic matching compares individual outputs using strategies including soft token overlap (e.g., BERTScore) and bipartite alignment. Soft token overlap computes token-level cosine similarity with contextual embeddings and aggregates soft precision, recall, and F1, giving partial credit for semantically similar but lexically distinct tokens (e.g., "physician" $\equiv$ "doctor") [163]. Bipartite matching enforces optimal one-to-one alignment between candidate and reference sets via the Hungarian algorithm, preventing double counting when multiple candidates map to the same reference [164].

Semantic Similarity Models. Semantic matching relies on dense embedding models:

- **Sentence-BERT (SBERT)** [141]: Optimizes BERT for semantic similarity tasks via siamese and triplet network structures with contrastive learning. SBERT produces fixed-size sentence embeddings suitable for efficient similarity search, achieving state-of-the-art results on semantic textual similarity (STS) benchmarks.

- **Universal Sentence Encoder (USE)** [165]: Captures semantic relationships through vector representations learned from large-scale corpora using transformer or deep averaging network architectures.

- **BGE (BAAI General Embedding)** [166]: An embedding model family achieving top performance on MTEB (Massive Text Embedding Benchmark), offering strong cross-domain generalization.

- **GTE (General Text Embeddings)** [167]: An embedding model family optimized for the expanded MTEB tasks, often used as strong off-the-shelf encoders for clustering and retrieval.

In addition, embeddings from model families such as Llama 3.1 and Qwen 2.5 are benchmarked on updated MTEB suites [151, 152, 168] and perform well on multilingual and semantic tasks. Compared to the specialized encoders discussed above, these LLM-derived embeddings are larger and more computationally demanding, making lightweight encoders easier to deploy and scale. This highlights a trade-off between benchmark performance and deployment efficiency.

While prior work has advanced semantic clustering with transformer embeddings, developed multi-LLM consensus methods with reliability measures, and compared similarity metrics across embedding models and granularities, these components have typically been evaluated in isolation. Chapter 5 integrates these components into a unified evaluation methodology that combines multi-model consensus with systematic validation across embedding representations and semantic granularities.

2.5 Synthesis and Research Gaps

This chapter has surveyed foundational concepts and prior work across three pillars. Building on this review, we now synthesize the remaining gaps. The following subsections outline each research question, highlight the specific methodological gaps identified in Sections 2.2–2.4, and preview how the corresponding contributions in Chapters 3–5 address them.

2.5.1 Representation: Domain-Specific Encoders

RQ1 (Chapter 3): Can adaptive loss-side weighting efficiently and robustly guide language models to *prioritize* domain-specific semantics and terminology beyond uniform-loss or data-only adaptation approaches?

Gap: Existing domain adaptation methods (Section 2.2) primarily focus on modifying input-side masking strategies or applying static token importance scores. Prior work has not explored loss-side token-level weighting that dynamically modulates loss contributions using adaptive signals. Moreover, stabilization of such weighted losses in masked language modeling has not been explicitly addressed.

→ **Chapter 3** introduces AMLM, which addresses these limitations through adaptive importance weighting while ensuring training stability.

2.5.2 Reasoning: Multi-Agent Coordination for QA Generation

RQ2 (Chapter 4): Which principles and mechanisms enable automated generation of QA pairs that are both *comprehensive* and *faithful* to technical context, particularly in settings requiring deep engagement and multifaceted reasoning?

Gap: Multi-agent reasoning systems (Section 2.3) tend to emphasize either diversity (e.g., ensemble decoding, debate) or convergence (e.g., self-consistency, critique), but lack mechanisms to simultaneously preserve perspectival diversity while ensuring semantic alignment. Prior work has not explored principled termination criteria that prevent both premature convergence and unnecessary computation in multi-agent reasoning processes.

→ **Chapter 4** introduces Collective Intentional Reading through Reflection and Refinement (CIR3), which addresses these challenges by balancing perspectival diversity with semantic alignment through structured multi-agent coordination and principled convergence mechanisms.

2.5.3 Evaluation: Consensus-Based Semantic Agreement

RQ3 (Chapter 5): How can we perform *rigorous*, *interpretable*, and *reliable* semantic evaluation of NLP systems in the absence of human-provided gold standards, leveraging semantic consensus and reliability estimation?

Gap: LLM-as-judge approaches (Section 2.4) lack mechanisms for assessing inter-model reliability. While semantic clustering and soft-matching methods exist, prior work has not integrated these approaches into a unified protocol for holistic reliability assessment.

→ **Chapter 5** introduces a consensus-based semantic evaluation framework that establishes reliability through multi-model agreement, employs complementary pairwise metrics, and validates stability across embedding representations.

2.5.4 Semantic Fidelity as Unifying Lens

The gaps identified above, while distinct in their technical manifestations, share a common thread: they reflect limitations in preserving semantic fidelity throughout the lifecycle of specialized domain NLP systems. Semantic fidelity (Chapter 1), the preservation of intended meaning and relations among domain concepts, provides a unifying perspective: (i) domain adaptation methods that treat all tokens uniformly lack mechanisms to preserve the semantic distinctions that define specialized knowledge (Section 2.2); (ii) multi-agent reasoning systems that emphasize either diversity or convergence, but not both, provide insufficient means to maintain semantic faithfulness while achieving comprehensive coverage (Section 2.3); and (iii) evaluation approaches that rely on single-model judgments or exact lexical matching offer limited capacity to assess semantic equivalence when meaning is preserved despite surface variation (Section 2.4).

Chapters 3–5 address these limitations through an integrated framework where semantic fidelity serves as the organizing principle, connecting representation learning, reasoning architectures, and evaluation methodologies for specialized domains.

Chapter 3

Adaptive Learning for Domain Specialization: A Framework for Modulated Masked Language Modeling in Finance

Abstract. *Specialized domains require models that capture domain-specific semantics, yet standard masked language modeling (MLM) assigns equal loss to all masked tokens, limiting emphasis on terminology that carries disproportionate semantic weight. We introduce Adaptive Masked Language Modeling (AMLM), a domain adaptation framework that applies token-level importance weights directly to the MLM loss function, shifting the adaptive learning paradigm from input-side masking to output-side gradient modulation. We instantiate four weighting strategies leveraging diverse contextual signals: sequence-level dynamics, batch composition awareness, corpus-wide rarity, and signal fusion. To ensure robust training under skewed weight distributions, we introduce stabilization mechanisms combining temperature smoothing and effective sample size targeting. Evaluated on financial domain tasks, AMLM substantially outperforms continued pre-training with standard MLM, improving QA matching Recall@1 by 0.231, semantic similarity by 0.188, and reducing intrinsic dimensionality from 23.762 to 9.847. By modulating token-level losses to prioritize domain-relevant terminology, AMLM improves representational semantic fidelity in specialized domains (Chapter 1), providing a complementary approach to existing input-side masking strategies and architectural adaptation methods.*

3.1 Introduction

Self-supervised pre-training, particularly through the Masked Language Modeling (MLM) objective [2], is a foundational approach in modern NLP. This paradigm has enabled the development of powerful, general-purpose language models like BERT and RoBERTa [24]. These models excel at a wide range of tasks by learning rich linguistic representations from vast, unlabeled text corpora [169]. However, their generalist nature often falls short in specialized domains such as finance [7], law [56], or biomedicine [6], where the precise understanding of technical terminology is critical for high-stakes applications. Consequently, adapting these foundational models to specialized domains remains an important and active area of research [3, 170].

The primary methods for domain adaptation involve either pre-training a model from scratch on a domain-specific corpus or continuing the pre-training of a general-purpose model on target-domain data. While training from scratch can yield highly specialized models, it is often prohibitively expensive in terms of data and computational resources [3]. Continued pre-training offers a more efficient alternative [6, 7], yet it is commonly implemented with the standard MLM objective, which treats all tokens with equal importance. This uniform treatment is a significant limitation: in financial text, domain-specific terms like "quantitative easing" or "collateralized debt obligation" represent a small fraction of tokens but carry disproportionate semantic weight for understanding. The model expends equal effort learning to predict common words as these critical domain concepts, leading to inefficient training and representations that are not fully attuned to the nuances of the specialized domain.

Existing research has attempted to address this limitation with adaptive training schemes [8, 9]. Many of these approaches, often categorized as "importance sampling", focus on modifying the model's input by developing more intelligent masking strategies [63, 25]. They use signals like token frequency or domain relevance to decide *what* tokens to mask, thereby concentrating the model's predictive efforts on more informative parts of the text. While effective, these methods primarily alter the distribution of the problem presented to the model, leaving the uniform loss computation unchanged.

In this work, we introduce a different and complementary paradigm: **Adaptive Masked Language Modeling (AMLM)**. Instead of modifying the token masking strategy (the input), AMLM operates on the model's output by introducing a knowledge-guided, weighted loss function. The central hypothesis is that by dynamically modulating the learning signal based on token importance, the model can be guided toward learning domain-aware representations. By amplifying the error signal for incorrect predictions on key terminology, the framework encourages the model to prioritize domain-specific concepts.

Specifically, this work addresses three key research questions:

1. **RQ1**: Can knowledge-guided loss weighting improve financial domain adaptation by prioritizing domain-specific terminology during pre-training?
2. **RQ2**: Which contextual signals (sequence dynamics, batch composition, corpus-wide rarity) are most suitable for identifying important domain-specific tokens during training?
3. **RQ3**: How can weighted loss training be stabilized to handle highly skewed importance distributions without compromising convergence?

To address these research questions, we make the following contributions:

1. We propose AMLM, a framework that applies importance weighting directly to the MLM loss function, shifting the focus of adaptive learning from input sampling to output-side gradient modulation (addressing **RQ1**).
2. We introduce four contextual weighting strategies that leverage diverse signals of token importance: sequence-level dynamics, batch composition awareness, corpus-wide rarity, and fusion of sequence and corpus signals (addressing **RQ2**).
3. We introduce gradient stabilization techniques, including temperature smoothing, Effective Sample Size (ESS) targeting, and weight clipping, to ensure robust training under highly skewed weight distributions (addressing **RQ3**).
4. We demonstrate substantial improvements on financial domain tasks while learning more compact representations.

The remainder of this work is organized as follows. Section 3.2 provides a detailed exposition of the AMLM framework, including the formulation of the loss function and the weighting strategies. Section 3.3 describes the experimental setup, Section 3.4 reports the results, followed by a concluding discussion.

3.2 Methodology

This research introduces an adaptive training scheme for masked language models that dynamically focuses on salient tokens during pre-training. Figure 3.1 provides a high-level conceptual overview of the AMLM framework. A detailed procedural flowchart is available in Appendix Figure A.1. This section details the theoretical underpinnings of the approach, from the baseline MLM to the proposed AMLM and its associated weighting strategies.

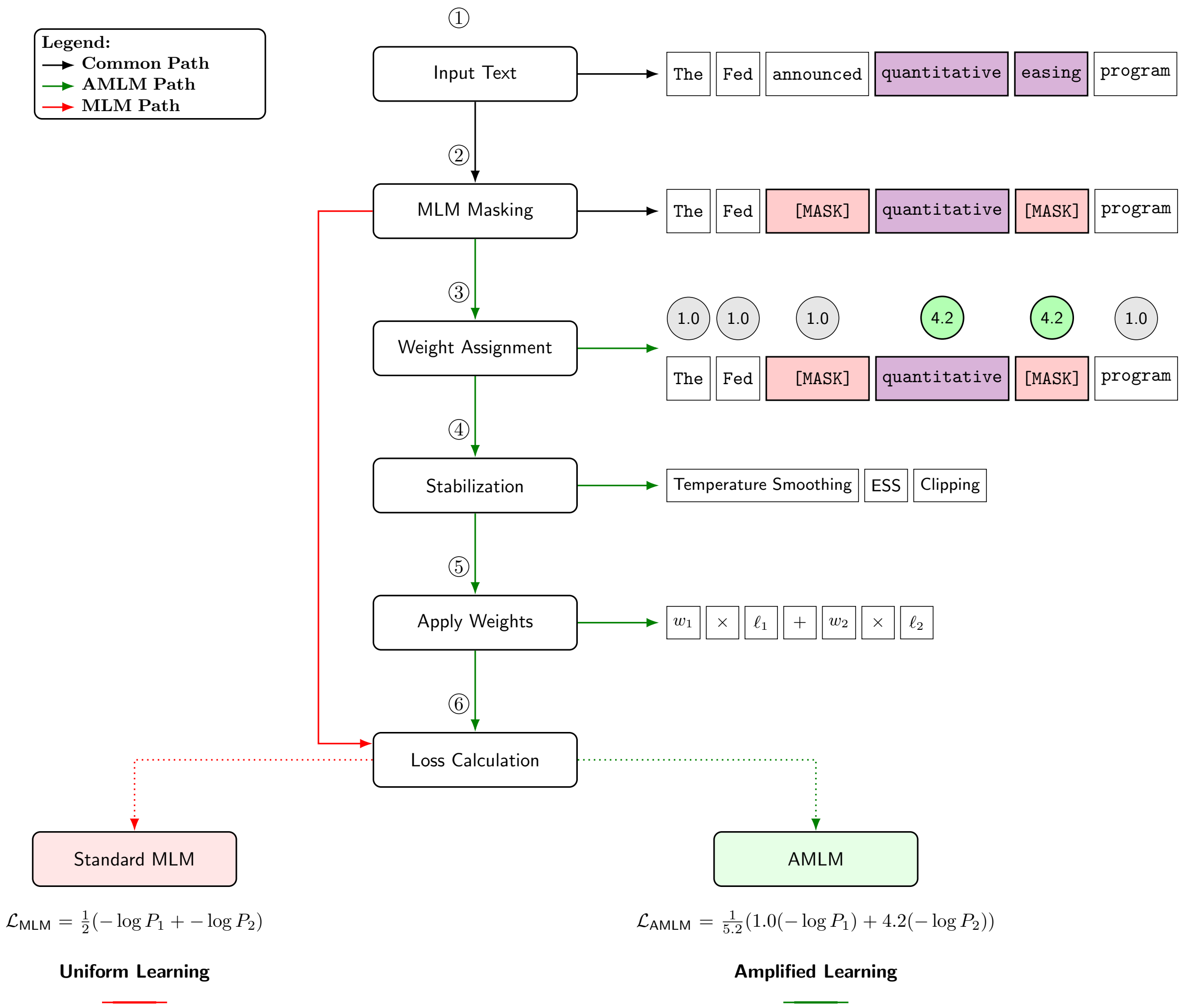

FIGURE 3.1: Overview of Adaptive Masked Language Modeling (AMLM). Both MLM and AMLM begin with input text (①) and standard masking (②). AMLM then assigns token-wise importance weights to domain jargon (violet tokens; ③), stabilizes these weights via temperature smoothing and ESS targeting (④), applies them to per-token losses (⑤), and aggregates the weighted losses (⑥). The MLM path (red) uses uniform, unweighted aggregation. Arrow colors denote flow types: common (black), AMLM-specific (green), and MLM-specific (red). Final arrows route the aggregated loss to the respective training objective boxes (Standard MLM in red, AMLM in green).

3.2.1 Foundation: Standard Masked Language Modeling (MLM)

This work is an extension of the MLM objective, which is the pre-training foundation for transformer-based models such as BERT [2]. The standard MLM process involves corrupting an input sequence $X = (x_1, x_2, \ldots, x_L)$ by randomly replacing a subset of its tokens with a special `[MASK]` token. The model is then trained to predict the original identities of these masked tokens based on the surrounding uncorrupted context.

Formally, for a set of masked indices M, the model minimizes the negative log-likelihood of the true tokens x_i conditioned on the unmasked context $X_{\setminus M}$:

$$L_{\text{MLM}}(\theta) = -\frac{1}{|M|} \sum_{i \in M} \log P(x_i | X_{\setminus M}; \theta) \tag{3.1}$$

where:

- θ represents the model parameters.
- $|M|$ is the number of masked tokens. Normalization by $|M|$ makes the loss comparable across sequences.
- x_i is the original token at masked position i.
- $P(x_i | X_{\setminus M}; \theta)$ is the predicted probability of the true token x_i.

This uniform-loss approach is limited in specialized domains, motivating the adaptive formulation presented next.

3.2.2 Adaptive Masked Language Modeling (AMLM)

A key limitation of the standard MLM objective is its uniform treatment of all tokens, where every masked token contributes equally to the loss function. However, not all tokens carry equal semantic importance. For instance, in financial reports, tokens representing economic indicators and financial concepts are more informative than common words. To address this, AMLM introduces a mechanism to differentially weight masked tokens. The central hypothesis is that by focusing the model on contextually or globally significant tokens, domain-aware representations can be learned more efficiently.

3.2.3 The AMLM Loss Function

AMLM is operationalized through a loss weighting mechanism that guides the model to focus on domain-specific tokens. This weighting amplifies the cross-entropy loss for domain-relevant tokens, increasing the penalty when the model fails to predict them correctly.

Formally, this is achieved by assigning a positive weight, $w_i > 0$, to each masked token x_i. The AMLM loss function is formulated as the weighted average of the per-token negative log-likelihood losses:

$$L_{\mathrm{AMLM}}(\theta) = -\frac{\sum_{i \in M} w_i \cdot \log P(x_i | X_{\setminus M}; \theta)}{\sum_{j \in M} w_j} \tag{3.2}$$

Normalized by the total weight of masked positions, the loss is a weighted average and thus invariant to uniform rescaling of w. Practical safeguards for rare edge cases follow the stabilized aggregation (Section 3.2.8).

The token probability is computed from the encoder logits via a softmax over the vocabulary:

$$P(x_i \mid X_{\setminus M}; \theta) = \frac{\exp\big(h_i^\top e(x_i)\big)}{\sum_{x' \in \mathcal{V}} \exp\big(h_i^\top e(x')\big)} \tag{3.3}$$

$$h_i = H_\theta(X_{\setminus M})_i, \tag{3.4}$$

where $H_\theta(X_{\setminus M})$ is the sequence of final hidden states from the encoder, h_i is the hidden state at masked position i, $e(\cdot)$ is the output embedding function, and $e(x')$ are embeddings for all tokens x' in the vocabulary $\mathcal{V}$. The output embeddings are tied to the input embeddings as in BERT.

As a simple instantiation that emphasizes a marked subset of masked tokens, the weight is set as:

$$w_i = 1 + \lambda s_i, \qquad s_i \in \{0, 1\}, \tag{3.5}$$

where s_i=1 if token i is marked by AMLM strategy and s_i=0 otherwise; λ>0 controls the amplification.

This formulation is a natural extension of the standard MLM loss and has two key properties:

1. **Modulated Gradient Contribution**: A token's contribution to the total loss, and therefore to the magnitude of the gradient updates during backpropagation, is directly proportional to its weight w_i. Higher weights signal greater importance.

2. **Scale Invariance**: By normalizing by the sum of the weights ($\sum w_i$) rather than the count of masked tokens ($|M|$), the loss function becomes invariant to the absolute scale of the weights. This ensures training stability and means the model learns from the *relative* importance of tokens, not their absolute weight values[1]. Section 3.2.8 introduces additional controls to further ensure stability against highly skewed weight distributions.

Having defined the conceptual basis of the AMLM objective, the following section specifies how to identify the domain-specific tokens that the weighting scheme will emphasize.

3.2.4 Domain-Specific Token Identification

AMLM requires accurately identifying domain-relevant tokens within the corpus. This work leverages a curated financial glossary to systematically locate and weight specialized terminology.

3.2.4.1 Glossary Construction and Encoding

Let $\mathcal{G} = \{g_1, g_2, \ldots, g_{|\mathcal{G}|}\}$ denote the financial glossary, constructed from FinRAD [171, 172] terms and containing specialized financial terminology such as "quantitative easing", "collateralized debt obligation", and "bull market." Given a subword tokenizer T (BERT tokenizer in this implementation) with vocabulary $\mathcal{V}$, an encoding function $\tau : \mathcal{G} \to \mathcal{V}^*$ is defined that maps each glossary term to its corresponding sequence of token IDs:

$$\tau(g_j) = (v_1^{(j)}, v_2^{(j)}, \ldots, v_{k_j}^{(j)}), \quad \text{where } v_r^{(j)} \in \mathcal{V} \tag{3.6}$$

Here, $k_j = |\tau(g_j)|$ represents the number of subword tokens required to encode term g_j.

3.2.4.2 Token-Level Jargon Matching

For a given input sequence $X = (x_1, x_2, \ldots, x_L)$ tokenized into token IDs $\mathbf{z} = (z_1, z_2, \ldots, z_{L'})$, jargon occurrences are identified through exact n-gram matching in token-ID space. The set of

[1]Normalization by $|M|$ or $\sum w_i$ ensures the MLM loss reflects a stable per-token metric, independent of the number of masked tokens. Weighted AMLM loss provides flexibility to emphasize specific tokens while preserving comparability across sequences. Standard MLM and weighted AMLM losses share the same underlying principle, with weights simply adjusting each token's contribution.

glossary matches is defined as:

$$\mathcal{I}(\mathbf{z}, \mathcal{G}) = \{(i, g_j) : \mathbf{z}_{[i,i+k_j)} = \tau(g_j), \text{ and } z_r \notin \mathcal{S} \text{ for } r \in [i, i + k_j)\} \tag{3.7}$$

where $\mathbf{z}_{[i,i+k_j)}$ denotes the subsequence of tokens from position i to $i + k_j - 1$, and $\mathcal{S}$ represents the set of special tokens (e.g., `[CLS]`, `[SEP]`, `[PAD]`) that are excluded from matching.

From the match set $\mathcal{I}(\mathbf{z}, \mathcal{G})$, a binary indicator function is derived to mark whether each token position belongs to any matched jargon term:

$$I_{\mathcal{G}}(i) = \begin{cases} 1, & \text{if } \exists (s, g_j) \in \mathcal{I}(\mathbf{z}, \mathcal{G}) \text{ such that } i \in [s, s + k_j) \\ 0, & \text{otherwise} \end{cases} \tag{3.8}$$

This indicator handles overlapping matches by taking the union of all covered positions, ensuring that each token position is marked at most once as jargon-related.

Algorithm 1 Domain Token Identification (Aho-Corasick) - implements the matching logic defined in Eqs. 3.7 and 3.8

Require:
Tokenized sequence $\mathbf{z} = (z_1, z_2, \ldots, z_{L'})$
Glossary $\mathcal{G}$
Encoding function τ
Special tokens $\mathcal{S}$
Optional: Pre-built Aho-Corasick automaton $\mathcal{A}$
Ensure: Domain token indicator $I_{\mathcal{G}}$, Match set $\mathcal{I}(\mathbf{z}, \mathcal{G})$
Initialize $\mathcal{I}(\mathbf{z}, \mathcal{G}) \leftarrow \emptyset$
Initialize $I_{\mathcal{G}}(i) \leftarrow 0$ for all $i \in [1, L']$
if automaton $\mathcal{A}$ is null or empty **then**
    Preprocessing: Build Aho-Corasick automaton from $\{\tau(g_j) : g_j \in \mathcal{G}\}$
end if
Scan: Traverse $\mathbf{z}$ with automaton $\mathcal{A}$, collecting matches
for each match (i, g_j) found by automaton **do**
    Get encoded term length $k_j = |\tau(g_j)|$
    if $z_r \notin \mathcal{S}$ for $r \in [i, i + k_j)$ **then**
        Add (i, g_j) to $\mathcal{I}(\mathbf{z}, \mathcal{G})$
        for $r = i$ to $i + k_j - 1$ **do**
            Set $I_{\mathcal{G}}(r) \leftarrow 1$
        end for
    end if
end for
return $I_{\mathcal{G}}$, $\mathcal{I}(\mathbf{z}, \mathcal{G})$

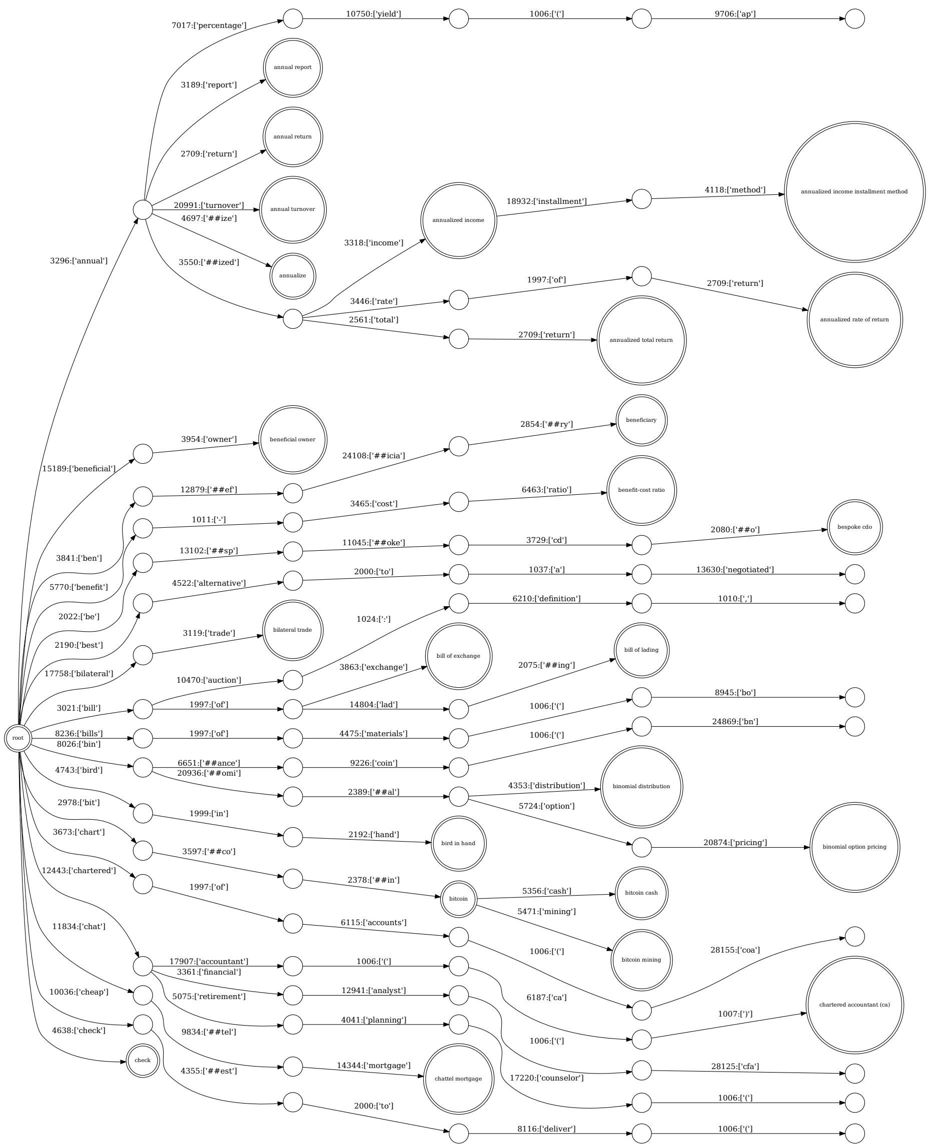

FIGURE 3.2: Example Aho-Corasick trie built from financial glossary terms (100 nodes shown; Max depth = 5). Nodes represent token IDs, edges show transitions. Terminal nodes (double circles) mark complete term matches. The trie enables linear-time multi-pattern matching over token-ID sequences.

The domain token identification process is formalized in Algorithm 1. While a naive nested-scan approach (Algorithm 5 in the Appendix) has complexity $\mathcal{O}(|\mathcal{G}| \times L' \times \bar{k})$, where $|\mathcal{G}|$ is the glossary size, L' the tokenized sequence length and $\bar{k} = \frac{\sum_j |\tau(g_j)|}{|\mathcal{G}|}$ is the average term length, our implementation uses the Aho-Corasick algorithm[2] [173] for efficient multi-pattern matching

[2]The Aho-Corasick algorithm is a classical, highly efficient method for searching multiple keywords (patterns) in a single pass over an input text, reporting all their occurrences simultaneously and in linear time relative to the text and pattern sizes. It is essential in NLP, compilers, DNA sequence analysis, and antivirus software, among other applications where fast dictionary-based matching is required.

over token-ID sequences. This enables a single-pass scan with complexity $\mathcal{O}(L' + \sum_j |\tau(g_j)|)$ after a one-time automaton construction.

The automaton is pre-built and cached from the tokenized glossary, preserving exact-match semantics while reducing preprocessing overhead. Figure 3.2 shows an example of the Aho-Corasick trie built from the financial glossary. The trie structure enables efficient multi-pattern matching by representing all glossary terms as paths through a single automaton. This visualization demonstrates how token-ID sequences are organized into a searchable tree structure.

3.2.4.3 Sequence-Level Jargon Statistics

For dynamic weighting strategies, aggregate jargon statistics are computed at both sequence and batch levels. For a sequence X with tokenized representation $\mathbf{z}$, the following definitions apply:

$$c_X = |\mathcal{I}(\mathbf{z}, \mathcal{G})| \quad \text{(jargon occurrence count, not constituent tokens)} \tag{3.9}$$

$$\rho_X = \frac{c_X}{|\mathbf{z}|} \quad \text{(jargon density)} \tag{3.10}$$

Note that c_X counts the number of occurrences of jargon terms, not the total number of tokens that comprise them. To avoid ambiguity from overlapping glossary terms, overlaps are resolved by retaining only the longest matching term at any given position. For example, if the text contains “collateralized debt obligation” and the glossary includes both “debt obligation” and “collateralized debt obligation”, the procedure counts this as a single match for the longest term, contributing $c_X = 1$.

For a training batch $B = \{X_1, X_2, \ldots, X_{|B|}\}$, the batch-level jargon density is computed as:

$$\rho_B = \frac{\sum_{X \in B} c_X}{\sum_{X \in B} |\mathbf{z}_X|} \tag{3.11}$$

These statistics enable the dynamic weighting strategies to adapt token importance based on local contextual information, as detailed in the following section. With these indicators and statistics in hand, they are converted next into explicit token weights w_i.

3.2.5 Contextual Weighting Strategies

The value of AMLM lies in its flexibility to accommodate various signals of token importance. Four distinct weighting strategies are formalized and evaluated against a standard MLM baseline. The core principle behind each strategy is to modify a baseline weight of $w_i = 1$ by applying a calculated "boost" to tokens identified as domain-specific jargon. This is operationalized using the indicator function $I_{\mathcal{G}}(i)$, which evaluates to 1 if the token at position i is part of a glossary term and 0 otherwise.

3.2.5.1 Baseline MLM

As a control, this strategy applies a uniform weight to all tokens, simplifying the AMLM loss (Equation 3.2) to the standard MLM loss (Equation 3.1).

$$w_i = 1, \quad \forall i \tag{3.12}$$

3.2.5.2 Self-Calibrating Contrast Weights

The base contrast between jargon and background tokens is set via two hyperparameters, α and β respectively, with $0 < \beta < \alpha$, both derived from the corpus-wide jargon density $\rho_{\mathcal{C}}$. The base contrast ratio is set to be inversely proportional to $\rho_{\mathcal{C}}$ to amplify the signal from rare jargon terms. However, a direct inverse relationship is overly aggressive for sparse distributions, so a self-calibrating dampening factor is introduced. The exponent is set equal to the corpus density itself, creating a dampening mechanism that adapts to the domain's jargon prevalence:

$$\frac{\alpha}{\beta} = \left(\frac{1}{\rho_{\mathcal{C}}}\right)^{\rho_{\mathcal{C}}} \tag{3.13}$$

A naive inverse scaling ($\alpha/\beta \propto 1/\rho_{\mathcal{C}}$) would explode the contrast as jargon becomes rare, over-amplifying noisy or ambiguous matches and destabilizing gradients. The self-calibrating dampening, where the exponent equals the corpus density, grows the contrast sublinearly in sparsity, preserving a clear emphasis for rare, domain-critical tokens while avoiding extreme weight ratios. This function $(1/x)^x$ has desirable limiting behavior: when jargon is sparse (small $\rho_{\mathcal{C}}$), the large base is strongly dampened by the small exponent; when jargon is abundant (large $\rho_{\mathcal{C}}$), dampening is less needed and the contrast naturally approaches unity. Practically, this improves the signal-to-noise trade-off in early training and reduces the likelihood of very low effective sample size under weighted aggregation[3] (Section 3.2.8).

Having established these base hyperparameters, the four adaptive weighting strategies are presented below, each incorporating different contextual signals anchored by α and β.

3.2.5.3 Sequence-Level Jargon Density (AMLM-Seq)

This strategy adapts weights based on the local context of each sequence by defining a dynamic boost factor that incorporates both jargon density ρ_X and a logarithmic scaling of the jargon occurrence count c_X. Let the boost factor be:

$$\text{boost}_X = 1 + \log(1 + c_X) \cdot \rho_X \tag{3.14}$$

When a sequence contains no jargon ($c_X = 0$), this factor is 1, creating a baseline weighting. As jargon density increases, the boost factor grows, amplifying the weights for jargon tokens and attenuating them for background tokens. This creates a dynamic contrast that focuses learning on jargon-rich contexts:

$$w_i = \begin{cases} \min(\alpha \cdot \text{boost}_X, w_{\max}) & \text{if } I_{\mathcal{G}}(i) = 1 \\ \max(\beta / \text{boost}_X, w_{\min}) & \text{otherwise} \end{cases} \tag{3.15}$$

where α and β (with $0 < \beta < \alpha$) are base weights defined from corpus density as in Section 3.2.5.2 (Equation 3.13).

The logarithmic scaling of the jargon count, $\log(1 + c_X)$, is crucial for several reasons. As a compressive function, it provides diminishing returns: while the first few jargon occurrences contribute significantly to the boost, additional occurrences provide progressively smaller increases. This prevents weight explosion in jargon-heavy sequences while still rewarding jargon density. For example, a sequence with 10 jargon occurrences receives a boost component of $\log(11) \approx 2.40$, while one with 100 occurrences gets $\log(101) \approx 4.61$, substantial but not linearly explosive.

[3] In the financial corpus, jargon constitutes 27.8% of tokens ($\rho_C \approx 0.278 \approx 1/4$), so the self-calibrating formula yields quarter-root dampening and a contrast ratio of $(1/0.278)^{0.278} \approx 1.38$. The parameter α is anchored at 0.9 (establishing jargon as a primary learning signal) so that the derived $\beta \approx 0.65$ remains moderate, ensuring background tokens still contribute. This value was validated through experiments on 30,000 sequences sampled uniformly at random with a fixed seed, where performance differences across $\alpha \in \{0.8, 0.9, 0.975\}$ were small.

The clipping bounds serve as additional safeguards. The hard ceiling at $w_{\max}$ prevents any outlier token from dominating the loss and causing gradient explosion. Conversely, the weight floor at $w_{\min}$ ensures that background tokens always contribute to the objective, preserving linguistic context and preventing vanishing gradients[4] for non-jargon terms.

In summary, the AMLM-Seq strategy implements three key mechanisms:

1. **Logarithmic scaling (`log1p`)**[5]: scales the effect of the jargon count sub-linearly via $\log(1 + c_X)$, damping extreme counts while preserving useful signal.
2. **Weight capping**: applies a hard ceiling with $\min(\cdot, w_{\max})$ to prevent any outlier from dominating the loss or causing gradient explosion.
3. **Weight flooring**: applies a lower bound with $\max(\cdot, w_{\min})$ so background tokens retain non-zero influence, preserving contextual learning and stable gradients.

3.2.5.4 Batch Composition (AMLM-Batch)

This strategy extends the dynamic approach by incorporating batch composition awareness to provide adaptive scaling across sequences with varying jargon distributions. The motivation is to maintain consistent signal strength when some batches contain predominantly jargon-rich sequences while others are jargon-sparse, which would otherwise lead to inconsistent gradient magnitudes across different batches.

Let σ_B be the standard deviation of jargon occurrence counts across all sequences in the batch. The square root of this standard deviation, $\sqrt{\sigma_B}$, serves as a composition-aware scaling factor that amplifies the dynamic effect based on batch heterogeneity while providing sublinear dampening to prevent excessive amplification in highly diverse batches. This transformation is analogous to the logarithmic dampening applied to jargon counts, ensuring that extreme batch heterogeneity does not destabilize training. The design choice reflects the principle that

[4] The *vanishing* and *exploding gradient* problems, first analyzed in recurrent networks [174, 175], can also affect deep Transformers. Large gradients can cause unstable updates (explosion), while small gradients can stall learning (vanishing). While architectural elements like *layer normalization*[73] and *residual connections*[72] provide some stability, extreme weight distributions in a loss function can still produce pathological gradients. A classic mitigation strategy is gradient norm clipping [176, 17]. Our use of weight clipping ($w_{\min}$, $w_{\max}$) is a related technique that directly bounds the contribution of any single token to the total loss, thus stabilizing the learning signal at its source.

[5] The `log1p` function, equivalent to $\log(1 + x)$, is used for enhanced numerical precision when its argument is close to zero.

higher batch-level variance indicates more heterogeneous content, warranting stronger adaptive differentiation to maintain consistent gradient signals across training steps.

The boost factor combines local sequence statistics with batch-level diversity:

$$\text{boost}_X = 1 + \log(1 + c_X) \cdot \rho_X \cdot \sqrt{\sigma_B} \tag{3.16}$$

The adaptive scaling mechanism works as follows. In uniform batches where all sequences have similar jargon counts ($\sigma_B \approx 0$), the $\sqrt{\sigma_B}$ factor approaches zero, causing the boost factor to approach 1 regardless of individual sequence statistics. This yields uniform baseline weighting with gentle differentiation. Conversely, in diverse batches containing both jargon-heavy and jargon-sparse sequences (high σ_B), the square root amplification becomes more pronounced, creating stronger differentiation to ensure that the diverse content receives appropriately scaled attention. This mechanism ensures that the full magnitude of the dynamic, sequence-level adjustments applies only in heterogeneous batches, stabilizing gradient magnitudes across different training steps.

Using this boost factor, the weights are computed as:

$$w_i = \begin{cases} \min(\alpha \cdot \text{boost}_X, w_{\max}) & \text{if } I_{\mathcal{G}}(i) = 1 \\ \max(\beta / \text{boost}_X, w_{\min}) & \text{otherwise} \end{cases} \tag{3.17}$$

where the clipping bounds (as defined in Section 3.2.5.3) ensure training stability.

This strategy's key components are:

1. **Dynamic foundation**: inherits all three stability controls from the dynamic strategy (logarithmic scaling, weight capping, weight flooring).

2. **Batch composition awareness**: computes σ_B to quantify the heterogeneity of jargon distributions within each batch during training.

3. **Composition-aware adaptive scaling**: uses $\sqrt{\sigma_B}$ to modulate the boost factor, providing stronger differentiation for diverse batches while remaining stable for uniform ones.

4. **Signal strength consistency**: automatically adjusts the weighting intensity to maintain consistent learning signals regardless of how random batching groups sequences together.

3.2.5.5 Corpus-Wide Rarity Using TF-IDF (AMLM-TFIDF)

This strategy adapts term frequency-inverse document frequency (TF-IDF), an established information retrieval method, to weight domain-specific tokens. The underlying principle is that terms appearing frequently within a document but rarely across the corpus possess greater discriminative value. For instance, a specialized term like "credit default swap" conveys more distinctive information than a generic term like "investment". Following the base weights α and β from Section 3.2.5.2, the weight is defined as:

$$w_i = \begin{cases} \min(\alpha \cdot (1 + \mathrm{TF}(t, X) \cdot \mathrm{IDF}(t)), w_{\max}) & \text{if } I_{\mathcal{G}}(i) = 1 \\ \beta & \text{otherwise} \end{cases} \tag{3.18}$$

where the components are:

$$\mathrm{TF}(t, X) = \frac{\mathrm{count}(t, X)}{|\mathbf{z}|} \quad \text{(local term frequency)} \tag{3.19}$$

$$\mathrm{IDF}(t) = \log\left(\frac{|\mathcal{D}|}{\mathrm{df}(t) + 1}\right) \quad \text{(global rarity)} \tag{3.20}$$

Here, $\mathrm{df}(t)$ is the document frequency of term t, $|\mathbf{z}|$ is the tokenized length of X, and the $+1$ smoothing prevents division by zero. The additive constant ("1+") in Equation 3.18 ensures that frequent but semantically important terms (e.g., "market") retain baseline weight α even when IDF is low. The clipping bound (as defined in Section 3.2.5.3) prevents extreme weights. Unlike the dynamic strategies, background tokens receive constant weight β as TF-IDF scoring focuses exclusively on term-level corpus statistics without sequence-level modulation.

The IDF scores are pre-calculated across the entire corpus during preprocessing and stored for efficient lookup, ensuring corpus-wide rarity statistics. TF scores are computed dynamically for each sequence during training.

The approach combines three key mechanisms:

1. **Local frequency weighting**: uses TF to reward terms that appear frequently within specific sequences, indicating topical relevance.

2. **Global rarity weighting**: uses IDF to substantially up-weight terms that are rare across the corpus, emphasizing specialized terminology.

3. **Multiplicative combination**: combines TF and IDF scores multiplicatively, creating the strongest weights for locally frequent but globally rare terms.

3.2.5.6 Fusion of Sequence and Corpus Signals (AMLM-Fusion)

This strategy implements *context-adaptive TF-IDF* weighting, where the standard TF-IDF signal is amplified by a sequence-level density factor (boost_X). The approach combines sequence-level context with term-specific importance, creating an adaptive baseline that responds to document technical density with fine-grained adjustments for individual terms.

The implementation follows a two-stage computational process. First, dynamic baseline weights are calculated as in the AMLM-Seq strategy, establishing a context-aware multiplier based on sequence-level jargon density. Let $\text{boost}_X = 1 + \log(1 + c_X) \cdot \rho_X$. Then, for each specific jargon term, this baseline is modulated by the term's TF-IDF score:

$$w_i = \min(\alpha \cdot \text{boost}_X \cdot (1 + \text{TF}(t, X) \cdot \text{IDF}(t)), w_{\max}), \quad \text{if } I_{\mathcal{G}}(i) = 1 \tag{3.21}$$

Background tokens receive the dynamic background weight:

$$w_{\text{background}} = \max(\beta / \text{boost}_X, w_{\min}) \tag{3.22}$$

While boost_X and TF both reflect properties of sequence X, they capture different aspects: boost_X measures overall jargon density as an aggregate signal across all jargon terms in the sequence, while TF measures a specific term's individual frequency. The multiplicative combination creates an adaptive dynamic range: technical documents exhibit stronger weight differentiation between rare and common terms, while casual documents exhibit more uniform weighting, reflecting the principle that term specificity matters more in specialized contexts.

This hierarchical approach creates four distinct weighting regimes based on sequence context (boost_X) and term rarity (IDF):

(1) Globally rare terms (high IDF) in jargon-dense sequences (high boost_X) receive maximum attention.

(2) Globally common terms (low IDF) in jargon-dense sequences (high boost_X) receive moderate amplification.

(3) Globally rare terms (high IDF) in jargon-sparse sequences (low boost_X) receive moderate TF-IDF weighting.

(4) Globally common terms (low IDF) in jargon-sparse sequences (low boost_X) receive minimal attention.

For example, a highly specialized term like "collateralized debt obligation" (high IDF) appearing in a technical financial document (high boost_X) would receive the product of its context amplification and global rarity signal, creating the strongest possible learning emphasis.

This fusion strategy relies on four key mechanisms:

1. **Hierarchical two-stage computation**: first establishes a sequence-level context multiplier (boost_X), then applies term-specific TF-IDF refinement for fine-grained control.
2. **Multiplicative amplification**: uses product rather than sum to create adaptive baselines where term importance is modulated by document technicality.
3. **Multi-granularity signals**: combines aggregate sequence-level jargon density with individual term-level frequency and corpus-wide rarity, maximizing differentiation.
4. **Computational complexity**: represents the most expensive strategy as it combines the sequence-level boost computation from AMLM-Seq with the per-term TF-IDF calculations from AMLM-TFIDF, providing the most nuanced weighting.

By comparing the performance of models trained with these different strategies, the analysis aims to uncover which signals of importance are most beneficial for learning high-quality representations in the target domain. To support these weighting schemes in practice, a training configuration is adopted that maximizes contextual coherence and stability.

3.2.6 Single Sequence Without Next Sentence Prediction

As demonstrated by [25], using single input sequences without the Next Sentence Prediction (NSP) objective yields superior performance compared to BERT's original two-segment approach. This design choice is particularly beneficial for domain adaptation for two key reasons: (1) single-sequence training leverages longer contexts, enabling better capture of domain-specific relationships and terminology, and (2) it eliminates noise in MLM predictions that can arise from potentially unrelated conditioning contexts across segment boundaries.

Given the technical and interconnected nature of financial discourse, where understanding often depends on extended context, this single-sequence approach is adopted. Training uses full-length sequences rather than two concatenated half-segments, maximizing the contextual information available for learning domain-specific representations. This configuration is especially advantageous for the AMLM weighting strategies, as they can operate on coherent, single-domain sequences where jargon terms appear in their natural linguistic and semantic contexts.

The following section details the practical implementation of the AMLM objective that realizes these weights efficiently and stably.

3.2.7 Weighted Loss Aggregation

To practically implement the AMLM loss function described in Equation 3.2 within modern deep learning frameworks such as PyTorch, a specific computational strategy is required. A direct application of the standard `CrossEntropyLoss` module is insufficient, as its default behavior is to compute the mean loss over all elements in a batch, which conflicts with the need to apply custom per-token weights.[6]

The implementation consists of two custom components: `AMLMDataCollator` for handling token weights during batch preparation, and `AMLMTrainer` for computing the weighted loss function. The training process follows a two-stage approach within each batch:

1. **Disaggregated Loss Calculation**: We first compute the raw, un-aggregated cross-entropy loss for every individual token in the batch. This is achieved by initializing the cross-entropy loss function with no reduction[7]. This configuration overrides the default averaging behavior and returns a loss tensor with dimensions `(batch_size, seq_len)`, where each element represents the loss for the corresponding token. The model's logits undergo tensor reshaping for efficient computation:

$$\mathbf{Z} \in \mathbb{R}^{B \times S \times V} \rightarrow \mathbf{Z}' \in \mathbb{R}^{(B \cdot S) \times V} \tag{3.23}$$

[6] https://docs.pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

[7] `CrossEntropyLoss(reduction='none')` instead of the default `'mean'`, which averages losses uniformly.

where B is batch size, S is sequence length, V is vocabulary size, and $\mathbf{Z}$ denotes the logits tensor. The logits are then reshaped back to align with the token weights. For tokens that were not masked (conventionally assigned a label of -100), the framework correctly assigns a loss of zero.

2. **Manual Weighted Aggregation**: With the per-token losses available, we perform a stabilized weighted aggregation to compute the final batch loss. The simple weighted average from Equation 3.2 provides the conceptual basis, but to ensure robust training in the presence of highly skewed weight distributions, we implement the full procedure described in Section 3.2.8. This involves applying temperature smoothing and ESS targeting to the raw weights before computing the final weighted sum of the per-token losses.

Removing the mean reduction from the loss unlocks the ability to move from a standard, uniform-loss paradigm to a more flexible, weighted-loss as in AMLM. The implementation provides the necessary granularity to apply fine-grained, adaptive pressure during training while maintaining computational efficiency through vectorized operations and careful tensor management. The procedure is agnostic to the particular choice of token weights w_i, allowing different weighting schemes to be integrated with the stabilization controls in Section 3.2.8.

Computational Overhead: AMLM introduces minimal overhead compared to standard MLM. The primary cost is a one-time preprocessing step for domain token identification, which is cached after the first run using the pre-built Aho-Corasick automaton. During training, the additional operations (weight retrieval, temperature smoothing, ESS computation) are $\mathcal{O}(|M|)$ where $|M|$ is the number of masked tokens per batch, making them negligible compared to the transformer forward pass which dominates at $\mathcal{O}(B \cdot S^2 \cdot d)$ for self-attention, where B is batch size, S is sequence length, and d is model dimension.

Memory Requirements: The main memory overhead comes from storing pre-computed token weights for each sequence in the dataset. For a dataset of N sequences with average length L, this requires $\mathcal{O}(N \times L)$ additional storage. For our dataset of 781,000 sequences with average length 320 tokens, this amounts to approximately 0.8GB when stored as FP16 (781,000 $\times$ 320 $\times$ 2 bytes), which is manageable for most modern training scenarios.

3.2.8 Stabilization Mechanisms for Weighted Loss Training

To ensure robust training under highly skewed weight distributions, we introduce two lightweight mechanisms: temperature smoothing [177] and Effective Sample Size[8] (ESS) targeting [178, 179]. These controls preserve the AMLM objective's scale invariance while mitigating excessive concentration of weight mass on a few tokens, thereby maintaining stable gradient magnitudes without compromising the adaptive weighting scheme.

3.2.8.1 Temperature Smoothing

After computing and clipping raw token weights $w_i \in [w_{\min}, w_{\max}]$, we apply a temperature transform[9] to reduce skew before normalization, a technique analogous to its use in knowledge distillation for softening probability distributions [77]:

$$w_i \leftarrow w_i^{1/\tau}, \quad \tau > 1 \tag{3.24}$$

Larger τ (e.g., $\tau \in [1.2, 2]$) flattens the distribution, and $\tau = 1$ recovers the original weights.

3.2.8.2 Effective Sample Size (ESS) Targeting

After temperature smoothing reduces tail skew, we enforce per-step stability by targeting the ESS over the masked tokens M in the current mini-batch.

Let $\bar{w}_i$ denote normalized weights and $N = |M|$ the number of masked tokens. We compute the ESS:

$$\bar{w}_i = \frac{w_i}{\sum_{j \in M} w_j}, \qquad S_2 = \sum_{i \in M} \bar{w}_i^2, \qquad \text{ESS} = \frac{1}{S_2} \tag{3.25}$$

The ESS can be intuitively understood as the number of unweighted samples an equivalent weighted sample represents. For example, if ESS = 5 for a batch of 100 masked tokens, the weighted sample is equivalent to having only 5 unweighted samples (the batch loss is dominated

[8] A classic metric from importance sampling.

[9] Temperature smoothing applies a monotonic, model-agnostic compression $w_i \leftarrow w_i^{1/\tau}$ to flatten target weights before normalization, preserving ranking and improving stability. Focal loss [68] instead reweights the per-token loss by $(1 - p_t)^\gamma$ based on model confidence, downweighting easy tokens and emphasizing hard ones. Thus, smoothing calibrates targets while focal concentrates learning. The two are complementary. AMLM's modular design permits optional extensions such as focal-style modulation, which would reweight tokens based on prediction confidence to further emphasize hard examples. See Appendix A.3 for discussion of how such confidence-based weighting could integrate with AMLM contextual strategies.

by a few highly-weighted tokens), indicating extreme concentration that increases gradient variance and can hinder stable convergence.

To prevent excessive concentration, we mix the normalized weights with the uniform distribution[10] to meet a minimum ESS target $E_{\min} = \kappa \cdot N$ with $\kappa \in [0.3, 0.5]$. Choosing κ in this range retains roughly 30-50% of the masked-token mass, balancing gradient variance reduction with the intended adaptive emphasis. If $\mathrm{ESS}(\bar{w}) \geq E_{\min}$, we retain $\bar{w}$ unchanged (i.e., λ=1); otherwise, we form the mixture:

$$\tilde{w}_i = (1-\lambda)\frac{1}{N} + \lambda\,\bar{w}_i, \quad \lambda \in [0,1] \tag{3.26}$$

The squared ℓ_2 norm of the mixture is:

$$\sum_i \tilde{w}_i^2 = \frac{1}{N} + \lambda^2\big(S_2 - \tfrac{1}{N}\big), \tag{3.27}$$

so enforcing $\mathrm{ESS}(\tilde{w}) \geq E_{\min}$ yields the bound:

$$\lambda \leq \sqrt{\frac{\frac{1}{E_{\min}} - \frac{1}{N}}{S_2 - \frac{1}{N}}} \tag{3.28}$$

When $\mathrm{ESS}(\bar{w}) < E_{\min}$, this upper bound lies in $[0,1]$, and setting λ to the bound attains $\mathrm{ESS}(\tilde{w}) = E_{\min}$. To alter the original weight distribution as little as possible, we select the maximal mixing factor λ that satisfies this bound, corresponding to the minimal intervention required to achieve the target ESS. When the original ESS already exceeds the target, the bound yields $\lambda \geq 1$ and we simply take λ=1 (no mixing).

With the final mixed weights $\tilde{w}$, the stabilized batch loss is computed as the weighted sum of per-token negative log-likelihoods ℓ_i over masked positions[11]:

$$L_{\mathrm{AMLM}}^{\mathrm{stable}}(\theta) = \sum_{i \in M} \tilde{w}_i\,\ell_i, \qquad \ell_i = -\log P(x_i \mid X_{\setminus M}; \theta) \tag{3.29}$$

This procedure replaces the original loss computation in Equation 3.2 with a more robust objective that maintains stable gradient magnitudes while allowing AMLM to emphasize salient

[10] This approach is conceptually similar to label smoothing [180], which mixes one-hot targets with a uniform distribution to regularize the model.

[11] Implementation safeguards: batches with no masked tokens ($|M|$=0) are skipped; if post-processed weights yield $\sum_{i \in M} w_i$=0, the sample is skipped; if the sum is positive but numerically tiny, an ε-guard is applied during normalization (e.g., 10^{-12} FP32, 10^{-6} FP16).

Algorithm 2 AMLM Training Loop with Stabilized Loss

Require:
Corpus $\mathcal{D}$
Glossary $\mathcal{G}$
Strategy $\mathcal{S}$
Model θ (initialized from pre-trained checkpoint)
Ensure: Trained model θ^*
Preprocessing Phase:
Encode glossary terms: $\tau(g_j)$ for all $g_j \in \mathcal{G}$
Calculate IDF scores: IDF $\leftarrow$ calculate_idf$(\mathcal{D}, \mathcal{G})$ (if needed)
for each sequence $X \in \mathcal{D}$ **do**
    Tokenize sequence: $\mathbf{z} \leftarrow \text{tokenize}(X)$
    Identify domain tokens: $\mathcal{I}, I_{\mathcal{G}} \leftarrow$ Algorithm 1$(\mathbf{z}, \mathcal{G}, \tau, \mathcal{S})$
    Calculate weights: $\mathbf{w} \leftarrow \mathcal{S}(\mathbf{z}, \mathcal{I}, I_{\mathcal{G}}, \text{IDF})$
    Store weights in cached dataset
end for
Training Phase:
for each training batch $\mathcal{B}$ **do**
    Apply MLM masking: $(\mathbf{X}_{\text{masked}}, \mathbf{Y}, \mathbf{M}) \leftarrow \text{mlm\_mask}(\mathcal{B})$
    Retrieve pre-computed weights: $\mathbf{W} \leftarrow \text{get\_weights}(\mathcal{B}, \mathcal{S})$
    Forward pass: $\mathbf{P} \leftarrow f_\theta(\mathbf{X}_{\text{masked}})$
    Compute per-token losses: $\boldsymbol{\ell} \leftarrow \text{CrossEntropy}(\mathbf{P}, \mathbf{Y}, \text{reduction='none'})$
    Stabilize weights: $\tilde{\mathbf{W}} \leftarrow \text{Stabilize}(\mathbf{W}, \tau, \kappa)$ (temperature smoothing + ESS targeting)
    Compute final loss: $\mathcal{L} \leftarrow \sum_i \tilde{w}_i \ell_i$
    Update model: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$ (with learning rate scheduling)
end for
return θ^*

tokens. The stabilization controls[12] jointly limit high-variance stochastic gradients while preserving relative importance: clipping bounds per-token influence, temperature smoothing compresses the weight distribution tail without destroying ordering, and ESS targeting guarantees a floor on effective sample size. Empirically this yields smoother loss traces and reduced generalization gaps (see Figure 3.4) without requiring smaller learning rates.

The training process then minimizes the expected value of this stabilized loss over the data distribution:

$$\min_\theta \mathbb{E}_{X,M}[L_{\text{AMLM}}^{\text{stable}}(\theta)] \tag{3.30}$$

where the expectation is taken over input sequences X and random masking patterns M. Optimization uses AdamW [181], with linear warmup followed by a linear decay learning rate schedule [16, 182]. The complete AMLM training procedure is formalized in Algorithm 2.

[12]Empirical analysis finds $\tau \in [1.2, 2.0]$, $\kappa \in [0.3, 0.5]$, $w_{\min}$=0.1, and $w_{\max}$=10 to be robust defaults. Larger τ and κ provide stronger smoothing when weight skew is extreme (e.g., jargon-dense batches). These controls retain AMLM's scale invariance while constraining distributional skew, stabilizing convergence without diluting emphasis on salient tokens.

3.3 Experiments

To evaluate the AMLM framework, experiments measure the quality of learned text embeddings. The evaluation spans intrinsic and extrinsic tasks probing semantic coherence, clustering quality, and QA matching. This section describes the dataset and preprocessing, the pre-training configuration, and the evaluation methodologies.

3.3.1 Training Dataset and Preprocessing

A specialized financial corpus was curated by combining four publicly available datasets:

1. **NASDAQ News** [183]: Financial news articles covering stock markets, companies, and economic events.
2. **Earnings Calls** [184]: Transcripts of quarterly earnings calls, including executive statements and analyst QA.
3. **Financial Alpaca** [185]: Short instruction-following texts and sentiment-labeled QA pairs from financial datasets.
4. **Finbro** [186]: Financial instruction-following dataset with concise explanations and examples.

3.3.1.1 Preprocessing and Segmentation

All texts were tokenized using the `bert-base-uncased` WordPiece tokenizer. Each document was split into segments using a sliding window[13] approach with the parameters detailed in Table 3.1. The dataset composition across different sources is shown in Table 3.2.

The financial glossary, used for identifying domain-specific terms, contains over 13,000 terms extracted from FinRAD[14] [171], covering a wide range of financial terminology from basic concepts (e.g., “interest rate”, “dividend”) to specialized instruments (e.g., “credit default swap”), ensuring broad coverage of domain-specific vocabulary.

[13] The sliding window ensures overlapping context for MLM and preserves multi-sentence reasoning.
[14] https://huggingface.co/datasets/sohomghosh/FinRAD_Financial_Readability_Assessment_Dataset

Table 3.1: Preprocessing Parameters and Statistics

Parameter / Statistic	**Value**
Segment length	128–512 tokens (avg 320)
Sliding window stride	128 tokens (∼40% overlap)
Total segments generated	781,000
Validation split	5% (39,050 segments)
Training split	95% (742,950 segments)
Total tokens (all segments)	∼249,920,000
Average tokens per segment	327

Table 3.2: Dataset Statistics and Segmentation

Source	**Segments Generated**	**Dataset Ratio**
NASDAQ News	400,000	0.51
Earnings Calls	250,000	0.32
Financial Alpaca	50,000	0.07
Finbro	80,000	0.10
Total (after sampling)	781,000	1.00

3.3.2 Experimental Setup

We compare the four proposed dynamic AMLM weighting strategies against a standard MLM baseline under a unified pre-training setup on the financial corpus. All models share identical data preprocessing, tokenization, and train/validation splits to ensure a controlled comparison.

3.3.2.1 Pre-training Configuration

All experiments initialize from the official `bert-base-uncased` checkpoint. We train five models in total: a standard `MLM Baseline` and four AMLM variants, as defined in Section 3.2.5: `AMLM-Seq`, `AMLM-Batch`, `AMLM-TFIDF`, and `AMLM-Fusion`.

The training configuration is kept constant across runs (Tables 3.3-3.4). We train for up to 11 epochs with early stopping on validation loss (patience 2) and report results from the checkpoint attaining the lowest validation loss (e.g., epoch 9). A single global seed (42) is used across Python/NumPy/PyTorch via `transformers.set_seed(42)`. cuDNN is set to deterministic mode (`deterministic=True`, `benchmark=False`), and DataLoader workers are seeded. Hardware/software specifications and training-time details appear in Appendix A.9. Reproducibility controls are summarized in Appendix A.10.

TABLE 3.3: Pre-training Hyperparameters

Parameter	**Value**
Base Model	`bert-base-uncased`
Optimizer	AdamW (betas $[0.9, 0.999]$, $\epsilon = 1 \times 10^{-8}$)
Learning Rate	2e-5
Learning Rate Schedule	Linear warmup (0 $\rightarrow$ 2e-5, 1,500 steps) Linear decay (2e-5 $\rightarrow$ 0)
Warmup Steps	1,500
Training Epochs	11 (with early stopping, patience 2 epochs)
Train Batch Size (per device)	16
Gradient Accumulation Steps	16
Effective Batch Size	256
Steps per Epoch	∼3,066
MLM Probability	0.15
Mixed Precision (FP16)	Yes
Weight Decay	0.01
Logging Steps	100
Evaluation Steps	500
Save Steps	1,000
Segment Length (tokens)	Variable: 128–512 (avg 320)
Sliding Window Stride	128 tokens (∼40% overlap)

TABLE 3.4: AMLM Weighting and Stabilization Hyperparameters. The value for β is derived via Equation 3.13.

Parameter	Value
Base jargon weight (α)	0.9
Background weight (β)	0.65
Weight floor (w_{min})	0.3
Weight cap (w_{max})	10.0
Temperature (τ)	1.5
ESS target ratio (κ)	0.4

We fix the tokenizer, corpus and splits, optimizer (AdamW), learning rate and schedule, batch sizing and accumulation, weight decay, mixed precision, and all RNG seeds. The sole experimental variable is the loss-weighting strategy.

3.3.3 Evaluation Protocol

We evaluate learned embeddings without downstream fine tuning to assess intrinsic and extrinsic semantic capabilities. Evaluations use the **FinLang/investopedia-embedding-dataset**[15], **TheGoldmanEncyclopedia**[16], and **SmoothNLPNews**[17]. For each model, we extract last

[15] https://huggingface.co/datasets/FinLang/investopedia-embedding-dataset
[16] https://huggingface.co/datasets/FinanceMTEB/TheGoldmanEncyclopedia-en
[17] https://huggingface.co/datasets/FinanceMTEB/SmoothNLPNews

layer embeddings with mean pooling[18]. Embeddings are L2 normalized and compared using cosine similarity. Stochastic procedures (e.g., k-means) are repeated three times with different seeds and reported as means. Other evaluations are deterministic for a fixed model.

3.3.3.1 Semantic Similarity

We assess semantic alignment on the 22,940 QA pairs from the Investopedia dataset, comparing model-generated similarity scores to reference scores from `bge-large-en-v1.5`[19], a strong baseline with strong cross-domain generalization. Performance is measured with correlation metrics (Pearson's r, Spearman's ρ, Kendall's τ) and calibration error metrics (MSE, MAE). Rank-based correlations are used to evaluate the monotonic agreement (whether a model correctly orders the similarity of pairs), which is a more robust indicator of semantic understanding than linear correlation alone. We report 95% confidence intervals to assess statistical significance, reflecting metric stability on the dataset. Metric definitions are provided in Appendix A.4. Chapter 5 further examines the alignment between AMLM-Fusion and BGE-Large as part of the cross-encoder robustness analysis.

3.3.3.2 Embedding Space Analysis

We analyze the intrinsic geometric properties of the learned representations on the 45,880 Investopedia embeddings. This analysis focuses on:

- **Intrinsic Dimensionality**: The estimated dimensionality of the embedding manifold (kNN-MLE), reflecting representational complexity. Lower values indicate more compressed, specialized representations [187].
- **Cluster Validity**: The natural clustering tendency of the space, measured via the Davies-Bouldin Index [188] (lower is better) and Calinski-Harabasz Score [189] (higher is better) without ground-truth labels.

Formal definitions are provided in Appendix A.5.

[18] We compare CLS, mean, and max pooling. Mean pooling performs best and is used for all reported results.
[19] https://huggingface.co/BAAI/bge-large-en-v1.5

3.3.3.3 Financial QA Matching

We evaluate retrieval performance on three question-answer matching benchmarks: **Investopedia**, **TheGoldmanEncyclopedia**, and **SmoothNLPNews**. For each query, the task is to retrieve its corresponding answer from the full answer corpus of that dataset. Performance is measured with standard information retrieval metrics:

- **Recall@K**: The fraction of queries where the correct answer is in the top-K results ($K \in \{1, 5, 10\}$).[20]
- **Mean Reciprocal Rank (MRR)**: The average reciprocal rank of the correct answer. In this single-positive setting, MRR is equivalent to Mean Average Precision (MAP).
- **nDCG@10**: Normalized Discounted Cumulative Gain at K=10, which rewards correct items ranked higher.

Formal metric definitions are provided in Appendix A.6.

3.4 Results

This section presents the empirical results of the experiments. Performance on semantic similarity is analyzed first, followed by an intrinsic analysis of the embedding space focusing on representational efficiency. The section concludes with the strongest results on the QA matching benchmarks, where the benefits of the AMLM framework are most pronounced.

The evaluation compares the four AMLM variants against a standard MLM baseline `BERT-MLM-CP` (Continued Pre-training). The results demonstrate that AMLM achieves notable improvements in semantic understanding and representational efficiency, with some trade-offs in geometric clustering properties that reflect the method's focus on semantic rather than geometric optimization.

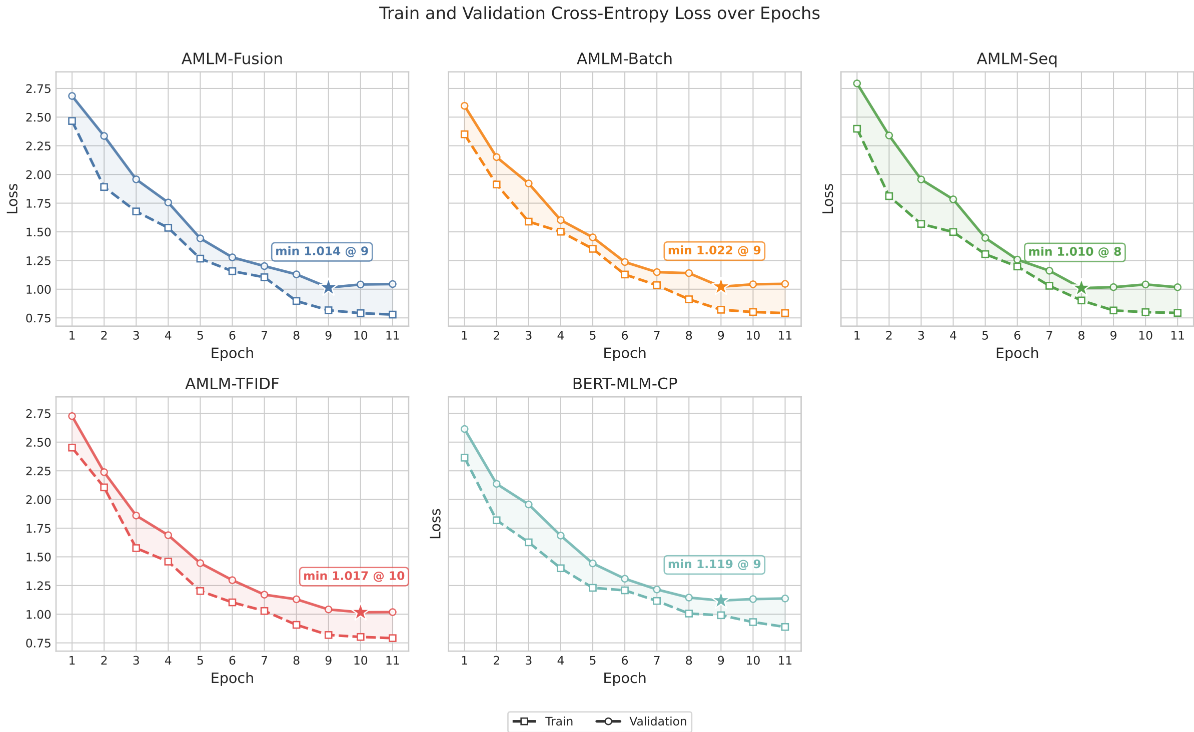

FIGURE 3.3: Training and validation cross-entropy loss over epochs for the AMLM variants and the `BERT-MLM-CP` baseline. Shaded areas represent generalization gaps. Annotated minima indicate selected checkpoints for evaluation.

3.4.1 Training Dynamics and Model Convergence

Figure 3.3 shows train and validation cross-entropy over 11 epochs for the `BERT-MLM-CP` baseline and four AMLM variants. All models converge stably with steadily decreasing losses. The AMLM variants, especially `AMLM-Fusion` and `AMLM-TFIDF`, reach lower validation minima and exhibit comparable or smaller generalization gaps than the baseline, indicating improved generalization. The annotated minima indicate the selected checkpoints for evaluation (e.g., `AMLM-TFIDF` 1.017 at epoch 10 vs. `BERT-MLM-CP` 1.119 at epoch 9). These dynamics motivate the downstream comparisons that follow.

In model comparison tables, the **best mean per metric** is shown in bold and the <u>second-best</u> is underlined. 95% confidence intervals are reported in brackets but not emphasized. Ablation tables report settings without emphasis.

[20] In a single-positive retrieval task, Precision@K is less informative, as it simplifies to $1/K$ for a successful retrieval and 0 otherwise. Recall@K directly measures the success rate and is therefore the more appropriate metric.

3.4.2 Semantic Similarity

As shown in Table 3.5, all AMLM variants notably outperform the MLM baseline on correlation-based metrics. The best-performing model, `AMLM-Fusion`, achieves the highest Pearson (0.914) and Spearman (0.873) correlations, reflecting stronger alignment with the reference ranking. While `AMLM-Seq` shows the best calibration with the lowest MSE (0.008) and MAE (0.072), the primary goal is capturing relative semantic ordering, for which correlation metrics are most informative.

Table 3.5: Semantic Similarity Evaluation Results (N=22,940 pairs). Correlations computed using mean pooling, L2 normalization, and cosine similarity. 95% confidence intervals reported in brackets. See Appendix A.4 for detailed methodology.

Model	**Pearson [95% CI]** ↑	**Spearman [95% CI]** ↑	**Kendall [95% CI]** ↑	**MSE** ↓	**MAE** ↓
`BERT-MLM-CP`	0.726 [0.720, 0.732]	0.650 [0.642, 0.658]	0.467 [0.460, 0.474]	0.011	0.081
`AMLM-Fusion`	**0.914** [0.912, 0.916]	**0.873** [0.869, 0.877]	**0.693** [0.689, 0.698]	0.014	0.082
`AMLM-Batch`	<u>0.892</u> [0.889, 0.894]	<u>0.858</u> [0.854, 0.862]	<u>0.671</u> [0.666, 0.675]	0.014	0.085
`AMLM-Seq`	0.879 [0.876, 0.882]	0.830 [0.825, 0.834]	0.640 [0.635, 0.645]	**0.008**	**0.072**
`AMLM-TFIDF`	0.869 [0.866, 0.872]	0.825 [0.820, 0.829]	0.633 [0.628, 0.637]	0.020	0.126

3.4.3 Embedding Space Analysis

The intrinsic geometric properties of the learned representations are analyzed to assess both efficiency and structural characteristics of domain-specific learning. An interesting finding (Table 3.6) is that AMLM reduces the manifold's intrinsic dimensionality (kNN-MLE) from 23.762 to 9.847 compared to the MLM baseline, indicating that the framework guides the model to learn more compressed and specialized representations of the financial domain.

Table 3.6: Embedding Space Analysis Results (N=45,880 embeddings). Embeddings computed via mean pooling. K-means uses K equal to the number of ground-truth topics. Intrinsic dimensionality estimated via kNN-MLE. Lower Davies–Bouldin and higher Calinski–Harabasz indicate better cluster structure.

Model	**Intrinsic Dim** ↓	**Davies–Bouldin** ↓	**Calinski–Harabasz** ↑
`BERT-MLM-CP`	23.762	**3.506**	**2583.864**
`AMLM-Fusion`	13.353	5.725	934.545
`AMLM-Batch`	13.643	5.350	1098.210
`AMLM-Seq`	<u>11.679</u>	5.091	1250.716
`AMLM-TFIDF`	**9.847**	5.695	1082.656

The cluster validity metrics reveal an interesting trade-off between manifold structure and cluster geometry. While the MLM baseline achieves tighter, more separated clusters (lower Davies–Bouldin, higher Calinski–Harabasz), AMLM learns more compact manifolds with lower intrinsic dimensionality. The reduction from 23.762 to 9.847 effective dimensions indicates that AMLM represents the financial domain in a lower-dimensional subspace. This manifold specialization comes at the cost of geometric cluster separation: AMLM prioritizes organizing representations by semantic similarity rather than forming tight, separated clusters. This trade-off aligns with AMLM's superior performance on semantic similarity and QA matching tasks, where semantic coherence matters more than cluster geometry.

3.4.4 Financial QA Matching

The financial question-answer matching task is where the benefits of AMLM are most pronounced. As shown in Tables 3.7–3.9, all AMLM variants consistently outperform the MLM baseline across all three benchmarks. On the Investopedia dataset, the `AMLM-Fusion` model achieves a Recall@1 of 0.424 and an MRR of 0.591, notably outperforming the MLM baseline.

Table 3.7: Financial QA Matching Results on the Investopedia dataset (N=22,940 QA pairs). Scores represent the ability to match questions to their corresponding answers. In this single-positive retrieval task, only Recall@k, MRR, and nDCG@10 are reported (Precision@k equals Recall@k; MAP equals MRR).

Model	**Recall@1 ↑**	**Recall@5 ↑**	**Recall@10 ↑**	**MRR ↑**	**nDCG@10 ↑**
`BERT-MLM-CP`	0.193 [0.188, 0.198]	0.433 [0.426, 0.439]	0.485 [0.479, 0.491]	0.308 [0.303, 0.313]	0.346 [0.341, 0.351]
`AMLM-Fusion`	**0.424** [0.417, 0.430]	**0.771** [0.766, 0.776]	**0.806** [0.801, 0.811]	**0.591** [0.586, 0.596]	**0.642** [0.637, 0.647]
`AMLM-Batch`	<u>0.402</u> [0.396, 0.408]	<u>0.767</u> [0.762, 0.773]	<u>0.802</u> [0.796, 0.807]	<u>0.577</u> [0.573, 0.582]	<u>0.631</u> [0.627, 0.636]
`AMLM-Seq`	0.385 [0.379, 0.391]	0.734 [0.728, 0.739]	0.769 [0.764, 0.774]	0.553 [0.548, 0.558]	0.605 [0.600, 0.610]
`AMLM-TFIDF`	0.391 [0.384, 0.397]	0.748 [0.742, 0.753]	0.782 [0.776, 0.787]	0.563 [0.558, 0.568]	0.615 [0.611, 0.620]

Table 3.8: Financial QA Matching Results on TheGoldmanEncyclopedia dataset (N=1,514 QA pairs). Scores represent the ability to match questions to their corresponding answers.

Model	**Recall@1 ↑**	**Recall@5 ↑**	**Recall@10 ↑**	**MRR ↑**
`BERT-MLM-CP`	0.025 [0.018, 0.033]	0.108 [0.092, 0.124]	0.151 [0.131, 0.170]	0.074 [0.064, 0.083]
`AMLM-Fusion`	**0.168** [0.149, 0.186]	**0.505** [0.480, 0.530]	**0.598** [0.571, 0.623]	**0.322** [0.304, 0.339]
`AMLM-Batch`	<u>0.143</u> [0.126, 0.161]	<u>0.399</u> [0.374, 0.424]	<u>0.500</u> [0.473, 0.525]	<u>0.265</u> [0.248, 0.283]
`AMLM-Seq`	0.127 [0.110, 0.144]	0.374 [0.350, 0.399]	0.482 [0.455, 0.507]	0.248 [0.232, 0.265]
`AMLM-TFIDF`	0.127 [0.110, 0.145]	0.375 [0.351, 0.400]	0.464 [0.438, 0.489]	0.244 [0.227, 0.260]

Table 3.9: Financial QA Matching Results on the SmoothNLPNews dataset (N=9,990 QA pairs). Scores represent the ability to match questions to their corresponding answers.

Model	**Recall@1 ↑**	**Recall@5 ↑**	**Recall@10 ↑**	**MRR ↑**
`BERT-MLM-CP`	0.000 [0.000, 0.001]	0.001 [0.000, 0.002]	0.003 [0.002, 0.004]	0.003 [0.003, 0.003]
AMLM-TFIDF	**0.050** [0.046, 0.054]	**0.161** [0.154, 0.168]	**0.203** [0.196, 0.211]	**0.105** [0.101, 0.110]
`AMLM-Fusion`	<u>0.045</u> [0.041, 0.049]	<u>0.145</u> [0.139, 0.152]	<u>0.188</u> [0.180, 0.196]	<u>0.097</u> [0.092, 0.101]
`AMLM-Batch`	0.010 [0.008, 0.012]	0.049 [0.045, 0.053]	0.083 [0.078, 0.089]	0.035 [0.032, 0.037]
`AMLM-Seq`	0.033 [0.030, 0.037]	0.102 [0.096, 0.108]	0.133 [0.126, 0.140]	0.070 [0.066, 0.074]

This performance indicates that AMLM models better capture the semantic relationships between financial questions and their corresponding answers, reflecting improved financial-domain understanding.

3.4.5 Ablation: Training Stabilization

To assess the impact of the stabilization mechanisms (temperature smoothing and ESS targeting), we conduct an ablation study on `AMLM-Fusion`, the best-performing strategy. For comparability, results are reported at each setting's best validation checkpoint (epoch 6 without stabilization; epoch 9 with stabilization). Stabilization substantially improves correlation-based semantic similarity metrics (Table 3.10) and QA matching effectiveness (Table 3.11).

TABLE 3.10: Effect of training stabilization on Semantic Similarity (N=22,940 pairs). Higher is better for correlations; lower is better for error metrics.

Setting	**Pearson ↑**	**Spearman ↑**	**Kendall ↑**	**MSE ↓**	**MAE ↓**
With Stabilization	0.914	0.873	0.693	0.014	0.082
Without Stabilization	0.733	0.660	0.475	0.010	0.078

TABLE 3.11: Effect of training stabilization on Financial QA Matching (Investopedia). Higher is better for all metrics.

Setting	**Recall@1 ↑**	**Recall@5 ↑**	**Recall@10 ↑**	**MRR ↑**	**nDCG@10 ↑**
With Stabilization	0.424	0.771	0.806	0.591	0.642
Without Stabilization	0.311	0.559	0.611	0.430	0.469

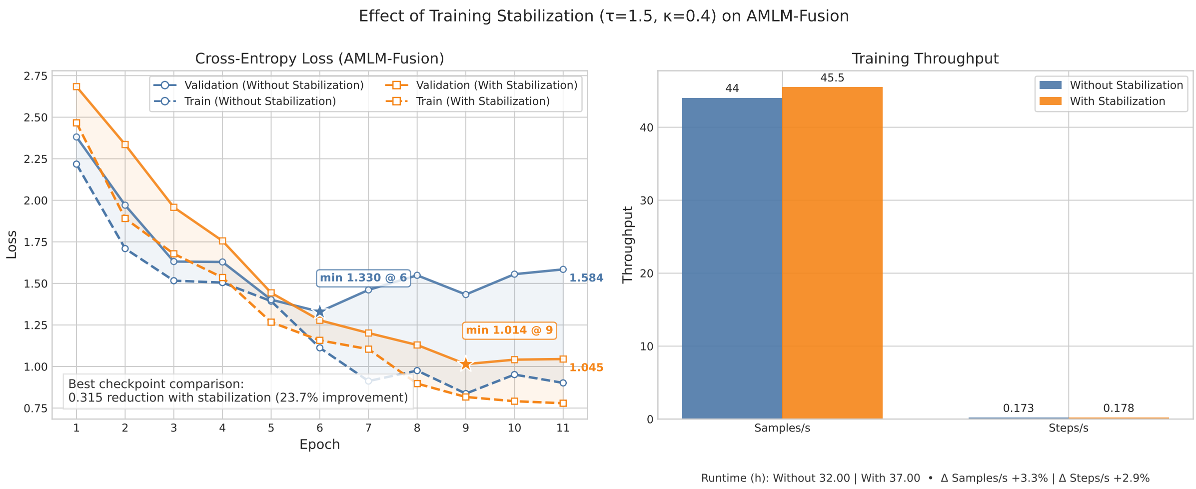

FIGURE 3.4: Training dynamics for `AMLM-Fusion` with vs. without stabilization. Left: train/-validation loss over epochs showing best checkpoints and generalization gaps. Right: throughput and runtime metrics. Stabilization reduces validation loss and generalization gap while maintaining efficiency.

Stabilization yields large, consistent gains. For semantic similarity, correlations increase notably (e.g., Pearson +0.181; Table 3.10), with a modest rise in calibration error (MSE/MAE). Because downstream retrieval emphasizes rank fidelity and our similarity labels are proxy-derived, we prioritize correlation gains and view the calibration trade-off as acceptable. For uncertainty reporting and methods, see Appendix A.4. On Investopedia QA matching, Recall@1 improves by +0.113 and MRR by +0.161 (Table 3.11), indicating sharper ranking of correct answers.

Figure 3.4 (τ=1.5, κ=0.4) compares the *best* validation checkpoints (epoch 6 without vs. epoch 9 with stabilization) and shows systematically lower validation loss and a reduced generalization gap under stabilization, mirroring the improvements observed in downstream metrics. Throughput is comparable across settings: samples/s and steps/s are similar, and total runtime differs marginally. Reported throughput values are single-run point estimates from the same runs that produced the loss curves (hardware in Appendix A.9).

3.5 Discussion

`AMLM-Fusion` exemplifies AMLM's core principle by combining local sequence context with global corpus rarity. The multiplicative fusion creates a product-of-experts effect: tokens that are both contextually salient and globally discriminative receive exponentially higher learning signals. This steers representations toward domain-critical subspaces, explaining the observed intrinsic dimensionality reduction without architectural changes.

Stabilization mechanisms ensure robust training under skewed weight distributions. Temperature smoothing compresses heavy-tailed distributions, while effective sample size targeting maintains stable gradient variance. Together, these controls preserve AMLM's adaptivity while preventing over-concentration on a few tokens.

The observed geometric trade-offs reveal a key characteristic of AMLM's representations. While manifold compression reduces dimensionality substantially, it does not produce tighter geometric clusters. Instead, AMLM organizes the lower-dimensional space to optimize semantic relationships rather than cluster separation. This trade-off aligns with AMLM's focus on semantic understanding rather than unsupervised clustering, explaining its superior performance on semantic similarity and QA matching tasks despite lower cluster validity scores.

3.6 Conclusion

We introduced Adaptive Masked Language Modeling (AMLM), a framework that enhances domain-specific pre-training by weighting token-level cross-entropy to prioritize domain-relevant terminology. AMLM achieves notable improvements in financial domain adaptation, with intrinsic dimensionality (kNN-MLE) reduced from 23.762 to 9.847, Pearson correlation improved by 0.188, and QA matching Recall@1 improved by 0.231, demonstrating that loss-side modulation effectively guides models toward specialized representations.

These improvements stem from AMLM's theoretical foundation: it reframes pre-training as importance weighted empirical risk minimization, where token weights w_i directly shape gradient contributions. Unlike input-side approaches that modify masking patterns or model architecture, AMLM preserves the standard MLM setup while selectively amplifying learning signals for domain-relevant tokens. This design makes AMLM orthogonal to and composable with such existing approaches (e.g., masking strategies and PEFT), opening new directions in domain adaptation and suggesting broader applicability to other specialized domains requiring targeted representation learning.

Within the dissertation's broader framework (Chapter 1), AMLM addresses representational semantic fidelity by learning embeddings that capture domain-specific semantics and conceptual relations. The observed improvements in semantic similarity and QA matching demonstrate that domain-adapted representations form a foundation for downstream generation and evaluation tasks in specialized domains. The robustness and generalizability of AMLM-Fusion as an embedding backbone is further validated in Chapter 5, where it is compared against general-purpose encoders in a consensus-based evaluation framework.

Future research will explore adaptive lexicon construction, cross-domain transfer, and integration with emerging pre-training paradigms. Key areas include unsupervised term identification, adaptive hyperparameter selection, and richer importance signals that combine syntactic and knowledge-informed approaches.

Chapter 4

Collective Reasoning: A Multi-Agent Framework for Faithful and Comprehensive QA Generation

Abstract. *Large Language Models (LLMs) excel at generating coherent and human-like questions and answers (QAs) across various topics, which can be utilized in various applications. However, their performance may be limited in domain-specific knowledge outside their training data, potentially resulting in low context recall or factual inconsistencies. This is particularly true in highly technical or specialized domains that require deep comprehension and reasoning beyond surface-level content. To address this, this chapter introduces* ***C****ollective* ***I****ntentional* ***R****eading through* ***R****eflection and* ***R****efinement (****CIR3****), a multi-agent framework that leverages collective intelligence for high quality Question-Answer Generation (QAG) from domain-specific documents. CIR3 employs a transactive reasoning mechanism to facilitate efficient communication and information flow among agents. This enables in-depth document analysis and the generation of comprehensive and faithful QAs. Additionally, multi-perspective assessment ensures that QAs are evaluated from various viewpoints, enhancing their quality and relevance. A balanced collective convergence process is employed to ensure that the agents reach a consensus on the generated QAs, preventing inconsistencies and improving overall coherence. Experiments indicate a substantial level of alignment between the CIR3-generated QAs and corresponding documents, while improving comprehensiveness by 23% and faithfulness by 17% compared to strong baseline approaches.*

4.1 Introduction

Question-Answer Generation (QAG) is a data augmentation task that consists of generating a set of QA pairs given a context. QAG has a variety of applications, from information retrieval [80, 81, 190] to healthcare [191, 192], and education [82, 83, 193]. Although Question Generation (QG) has been extensively researched in the context of language models [194, 195], QAG presents a more challenging task, as it requires generating both the question and the answer, rather than assuming that the answer is already provided in the input, as illustrated in Example 4.1. While QG models offer a more direct and focused approach, they primarily focus on surface-level features of the context, such as facts and keywords. This is due to the limited amount of explicit information that is conditioned on the input answer. Furthermore, despite the proposal of various methods, generating comprehensive and semantically distinct questions from the same context remains under-explored as highlighted in [93, 94]. The latter attributes this limitation to the lack of multi-reference training datasets that exhaustively cover all possible questions for each context. This inability is even more evident in highly technical or specialized domains, where documents are often rich in information. In the context of semantic fidelity (Chapter 1), this work addresses the generation stage by producing outputs that are both comprehensive and faithful while preserving domain-specific meanings.

Illustrative Example: QG vs. QAG

Context: *"A defined contribution pension plan is one where the final benefit depends on the contributions made and the performance of the selected investments."*

QG Output:

- How does a defined contribution plan work?
- What determines the final benefit in a defined contribution plan?

QAG Output:

- **Q:** How does a defined contribution plan work?
  A: It depends on the contributions and investment performance.
- **Q:** What determines the final benefit in a defined contribution plan?
  A: The final benefit depends on contributions and investment returns.

In-Context Learning (ICL) [1] is an emerging paradigm that enables LLMs to learn new tasks without the need for extensive fine-tuning on specific data. By providing a description of the task, along with a few or even zero demonstrations as part of the input context, LLMs can be conditioned to perform well in various domains. This approach has shown promising results, surpassing state-of-the-art models in some tasks, and offers a potential solution to the challenge of limited data availability [88, 196, 197]. Despite impressive results on popular NLP benchmarks, using ICL for QAG, given a relatively complex document, often lacks robust inference mechanisms to deduce implicit relationships between the different key points inherent in the context. If the generation depends on comprehending the underlying connections that are not explicitly stated in the context, the model may fail to generate faithful QAs that accurately reflect this complexity. This is particularly problematic for information-dense contexts, which are common in highly domain-specific corpora, such as finance and health.

Advancements in LLM-based Multi-Agent[1] (LLM-MA) systems have shown significant improvements in problem-solving abilities through planning, collaboration, and autonomous task execution [10, 11]. These systems break down complex tasks into simpler subtasks to enhance complex task solving. Compared to standard LLMs and single-agent setups, LLM-MA systems offer advanced capabilities by leveraging collective intelligence and specialized skills [198]. Motivated by the potential of these capabilities, the QAG task is augmented with collective reasoning through the adoption of LLM-MA settings.

To address the aforementioned limitations in relation to generating comprehensive and faithful QAs from highly domain-specific documents, this work derives a set of research questions around the adoption of LLM agents for QAG tasks: **(R1)** Can an LLM-MA workflow uncover deeper and perhaps implicit key concepts from a complex and information-dense document? **(R2)** How can LLM-MA effectively emphasize deep engagement with a text from different viewpoints to enable comprehensive and consistent generation and mitigate blind spots? **R3** (a) How can multiple agents be incentivized to seek consensus? (b) How can the convergence process be controlled to reach common QAG while avoiding premature collapse to incomprehensive and/or unfaithful generation?

To address these research questions, Collective Intentional Reading through Reflection and Refinement (**CIR3**) is designed based on three corresponding hypotheses:

[1]LLM-based agents are autonomous systems that leverage LLMs as their core reasoning and decision-making engine. These agents can perceive their environment through natural language, process information, generate plans, and take actions to achieve specific goals. Unlike traditional AI systems with static functionalities, LLM-based agents exhibit a degree of general intelligence, enabling them to handle a wider range of tasks and adapt to novel situations based on their extensive knowledge and language understanding capabilities[100, 150].

H.1 : Transactive reasoning[2] allows the deduction of QAs that uncover the implicit relationships between key concepts within the text.

H.2 : Multi-perspective group debate leads to an in-depth analysis of the document.

H.3 : Collective convergence, the process of a group of agents moving towards a shared output, requires disruptive signals to ensure diversity is maintained and collapse is avoided.

To build upon these hypotheses, CIR3 first utilizes an optimized topology of information within the agents to maximize the effectiveness of collaborative problem-solving and ensure an in-depth analysis of the input document. Second, CIR3 gains effectiveness by dynamically allocating specialized *writer* agents, each with a distinct perspective, based on the topic categories identified within the input context. Third, to reach a shared understanding of the document, despite the diverse perspectives and reasoning capabilities of the writers, CIR3 employs a *curmudgeon* agent as a mechanism for introducing variation. The curmudgeon, coupled with an external evaluation tool, incites the writers towards a balanced collective convergence on the key concepts within the text while maintaining diversity in the generated QAs.

While lexical matching is a standard evaluation method for QA tasks, its limitations become apparent when dealing with generative models, which often produce plausible answers not found in the predefined gold standard. This issue is further compounded by LLMs generating increasingly complex and lengthy answers, making lexical matching even less effective [201]. To ensure a comprehensive and accurate evaluation of CIR3, diverse automatic metrics are employed, in addition to human evaluation.

In summary, the main contributions include:

(1) Introduction of CIR3, a dual-loop multi-agent framework for QAG that formalizes Comprehensiveness and Faithfulness as a diversity-alignment objective and operationalizes transactive memory via inner/outer iterative refinement with explicit termination conditions.

(2) Design of a curmudgeon-guided convergence mechanism that maintains diversity while driving consensus, with systematic ablations isolating the contributions of agent reasoning versus diversity measurement.

[2] This work adapts the concept of transactive reasoning [199, 200], a cognitive process that occurs through social interaction, where individuals build upon each other's ideas to create new knowledge or solve problems. It involves a dynamic exchange of thoughts, critiques, and elaborations, leading to a deeper understanding of a topic.

(3) Demonstration of consistent improvements over strong LLM baselines across finance and medical domains through comprehensive automatic and human evaluations, with open-source implementation for reproducibility.

4.2 Method

Given a context c consisting of a text passage, the task of QAG aims to produce a set of QA pairs, denoted as $\mathcal{G} = \{(q_i, a_i)\}_{i=1}^{N}$, that satisfies two crucial properties:

1. **Comprehensiveness:** The set $\mathcal{G}$ should cover all the key points and essential information present in the context c. In other words, for every significant aspect or piece of information $x \in c$, there exists at least one QA pair $(q_i, a_i) \in \mathcal{G}$ such that q_i elicits and a_i provides information relevant to x.

2. **Faithfulness:** Each answer a_i in $\mathcal{G}$ must be grounded in and supported by the factual content of the context c. This constraint ensures that the generated answers are not fabricated or hallucinatory, but rather reflect accurate information derived from the given text.

Formally, the QAG task can be formulated as an optimization problem, where the objective is to find the set $\mathcal{G}$ that maximizes both comprehensiveness and faithfulness with respect to the context c. This can be expressed as: $\mathcal{G}^* = \arg\max_{\mathcal{G}} [\mathrm{Comp}(\mathcal{G}, c) + \mathrm{Faith}(\mathcal{G}, c)]$, where $\mathrm{Comp}(\mathcal{G}, c)$ and $\mathrm{Faith}(\mathcal{G}, c)$ are scoring functions that assess the extent to which the set $\mathcal{G}$ covers the key points of c and adheres to the factual content of c, respectively. These scoring functions are defined in terms of diversity measures as follows:

$$\mathcal{G}^* = \arg\max_{\mathcal{G}} \left[\underbrace{\frac{\alpha_{q,a}}{2} \cdot (D_q + D_a)}_{\mathrm{Comp}(\mathcal{G},c)} + \underbrace{\alpha_{a,c} \cdot (1 - D_{a,c})}_{\mathrm{Faith}(\mathcal{G},c)} \right] \tag{4.1}$$

where $\mathcal{D}_{\mathrm{q}}$ and $\mathcal{D}_{\mathrm{a}}$ denote diversity scores computed over the sets of generated questions $\{q_i\}_{i=1}^{N}$ and answers $\{a_i\}_{i=1}^{N}$, respectively, and $\mathcal{D}_{\mathrm{a},c}$ denotes a dissimilarity measure between the concatenated answers $a_1 \oplus \cdots \oplus a_N$ and the context c. $\mathcal{D} \in [1, 2] \subset \mathbb{R}; \mathcal{D} = 1$ denotes perfect similarity.

The coefficients[3] $\alpha_{q,a}$ and $\alpha_{a,c}$, where $\alpha_{q,a} + \alpha_{a,c} = 1$, control the relative weighting of question and answer diversity (Comprehensiveness) and the alignment of answers with the context (Faithfulness) in the overall score.

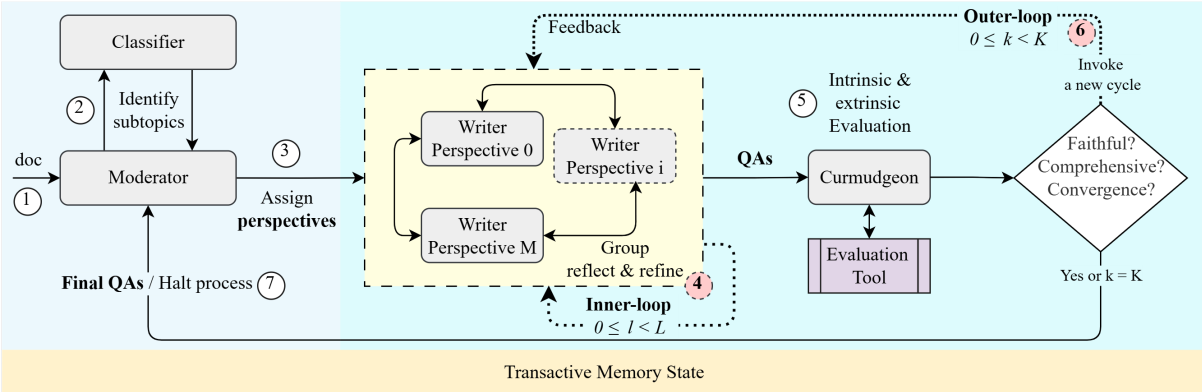

FIGURE 4.1: CIR3 takes an input document (1), identifies subtopics (2), and prompts writer agents to generate QA pairs based on their assigned perspectives (subtopics) (3). The QAs undergo iterative refinement by the writers (4), followed by an outer refinement where the curmudgeon, using its intrinsic knowledge and the evaluation tool, analyses the QAs and provides feedback for the next cycle (5, 6). The process halts when the curmudgeon is satisfied, and CIR3 returns the final QAs (7). The transactive memory serves as a central knowledge repository.

In what follows, CIR3 is described to generate the optimal solution $\mathcal{G}^*$ given c. This is achieved by building upon the aforementioned hypotheses to ensure that QAG is based on an in-depth analysis of the input text through an efficient flow of information and adoption of multiple views approach (4.2.1, 4.2.2), while maintaining QAG diversity and optimizing the convergence rate of agents (4.2.3). The pseudo-code of the algorithm serving as the conceptual foundation of the approach is outlined in Algorithm 3.

4.2.1 Multi-Perspective Analysis

Incorporating multi-perspective or various viewpoints is crucial for analyzing complex documents as it enhances the depth and breadth of understanding. Existing research highlights that a single perspective may introduce bias or overlook crucial aspects [202]. For instance, STORM [203] emphasizes the value of multiple perspectives in writing Wikipedia-like articles, by guiding participants to ask more in-depth questions in the pre-writing stage. Similarly, [204] showcased how addressing various perspectives improved document clarity and readability in document revision task.

[3]In this study, the coefficients $\alpha_{q,a}$ and $\alpha_{a,c}$ are empirically assigned equal weights (0.5). Although this choice effectively demonstrates the framework's capabilities, future research will explore dynamic estimation of α-values, potentially leveraging neural networks or other adaptive techniques, to further optimize Comprehensiveness and Faithfulness.

Algorithm 3 Pseudo-algorithm describing the CIR3 conceptual framework. The loops are designed to terminate gracefully when either the feedback is empty (indicating satisfactory) or the number of iterations reaches the predefined threshold. Further implementation details are given in B.2: Algorithm Implementation Details.

Require:
Max inner-refinement cycles L
Max outer-refinement cycles K
Max perspective M, Context c
Ensure: Refined QA pairs $\mathcal{G}^*$
Initialize: $\mathcal{M} \leftarrow$ [””] ▷ Writer's short memory state. (Eq. 4.2)
Initialize: $\mathcal{H} \leftarrow$ [””] ▷ Long short-term memory state. (Eq. 4.3)
// Identify and assign unique perspectives $\mathcal{P}$.
W_{P_0} = default in-domain writer
$W \leftarrow [W_{P_0}]$ ▷ List of Writers.
$P \leftarrow$ classify_subtopics(c, M) ▷ List of subtopics $\leq M$.
for $i = 1$ to $|P|$ **do**
    W.append(get_perspective_writer($P[i]$))
end for
Outer-refinement cycles: $k \leftarrow 0$; $\mathcal{F}\prime_{k+1} \leftarrow \varnothing$
repeat
    Inner-refinement cycles: $l \leftarrow 0$; $\mathcal{F}_{l+1} \leftarrow \varnothing$
    repeat
        ${\mathcal{G}^+}_l \leftarrow$ generate_QAs$(c, \mathcal{M}[-1], \mathcal{H}[-1])$
        $\mathcal{F}_{l+1} \leftarrow$ refine_QAs$(({\mathcal{G}^+}_l, \mathcal{M}))$
        $\mathcal{M}$.append$(({\mathcal{G}^+}_l, \mathcal{F}_{l+1}))$
        $l \leftarrow l + 1$
    until $l \geq L \vee \mathcal{F}_{l+1} = \varnothing$
    ${\mathcal{G}^-}_k \leftarrow {\mathcal{G}^+}_{l-1}$
    $\mathcal{F}\prime_{k+1} \leftarrow$ curmudgeon_QAs$(({\mathcal{G}^-}_k, \mathcal{H}))$
    $\mathcal{H}$.append$(({\mathcal{G}^-}_k, \mathcal{F}\prime_{k+1}))$
    $k \leftarrow k + 1$
until $k \geq K \vee \mathcal{F}\prime_{k+1} = \varnothing$
$\mathcal{G}^* \leftarrow {\mathcal{G}^-}_{k-1}$
return $\mathcal{G}^*$

While STORM efficiently identifies different perspectives by surveying existing articles from similar topics using a search engine, the CIR3 challenge is to discover diverse perspectives from a contained and limited context without retrieving external information. Given the input context c, CIR3 leverages LLM language understanding capabilities to identify different subtopics within the input document c. To this end, few-shot prompting, with a limited set of demonstrations, is employed to guide a *classifier* agent to classify the context into M specific categories $P = \{p_1, ..., p_M\}$ (Figure 4.1 ①-②)[4]. Next, the *moderator* agent dynamically assigns each identified perspective p_j to a different writer W_{p_j}, while prompting the agents to analyze

[4]For example, given a finance-related document, CIR3 is prompted to discover the different M subtopics present in the context, such as *pensions*, *insurance*, and *savings*.

the input context and generate a set of QA pairs, $\mathcal{G}^{p_j}$, based on their respective perspectives (Figure 4.1 ③). Subsequently, as per sections 4.2.2 and 4.2.3, the list of $\mathcal{G}^{p_j}$ are aggregated into $\mathcal{Q}^+ = \{\mathcal{G}^{p_j}\}_{j=1}^{M}$, then subjected to iterative refinement and evaluation, ultimately resulting in $\mathcal{G}^*$. For better coverage of the overall information and the relationships between the key concepts within the context, CIR3 introduces W_{p_0} based on the corpus domain. Additionally, this approach guarantees at least one agent will be available even if no subtopics are identified.

4.2.2 Transactive Reasoning

[205] explores how the group structure, the pattern of connections between individuals, can significantly influence collective cognition and shared knowledge within the group. This suggests that the structure of a network plays a crucial role in how memories are shared and aligned within a group. For instance, centralized networks, where information flows through a few key individuals, can lead to faster memory alignment but may also result in the loss of some details. In contrast, decentralized networks, with more diverse connections, may preserve a wider range of memories but take longer to reach consensus.

Drawing upon these insights, CIR3 employs a hybrid topology that consists of decentralized network of writer agents within a centralized network of two more agents, *moderator* and *curmudgeon*. To encourage communication and interaction, the group of writers form a fully-connected graph, where they operate at the same hierarchical level. To facilitate transactive reasoning, CIR3 adopts a reflection process, which benefits from the iterative exchange of critiques and refinements among the writers (Figure 4.1 ④). At iteration l, CIR3 gathers and aggregates feedback from all writers into $\mathcal{F}_l = \{\mathcal{F}_l{}^j\}_{j=1}^{M}$, links it to the previous QAs, $\mathcal{G}^+{}_{l-1}$, and then appends this updated information to the transactive memory. This creates a sequential memory state that evolves with each iteration:

$$\mathcal{M} = \{(\mathcal{G}^+{}_0, \mathcal{F}_1), ..., (\mathcal{G}^+{}_{l-1}, \mathcal{F}_l)\}_{l=0}^{L} \tag{4.2}$$

The reflection prompt is specifically designed to encourage the participants to build upon each other's analysis, while maintaining comprehensive and faithful output. To incentivize the agents to seek an optimal consensus, CIR3 builds upon the group's decentralized graph to (1) capitalize on the strengths inherent in centralized networks, and (2) incite the group towards a shared and optimal solution (4.2.3).

4.2.3 Guiding Collective Cognitive Convergence

In addressing R3, this work draws inspiration from the phenomenon of *Collective Cognitive Convergence (C3)* [206, 207] and from *How social network topology can shape collective cognition* [205].

4.2.3.1 Collective Cognitive Convergence (C3)

C3 [206, 207] is a sociology and evolutionary-biology related phenomenon that occurs when frequent interactions among a group lead to a convergence in cognitive[5] orientation. This phenomenon is observed in various contexts, including research sub-disciplines, political and religious associations, and persistent adversarial configurations. The authors highlight that while convergence facilitates mutual understanding and coordination, if left unchecked, it can lead to cognitive collapse, by reducing the diversity of concepts to which the group is exposed. Hence, limiting the group's ability to explore other viewpoints and generate new ideas. C3 suggests that the lack of mechanisms to introduce variation incites the collapse and reduction in diversity.

4.2.3.2 Social Network Topology

[209, 210] investigates how group characteristics, including composition and structure, explain variations in their Collective Intelligence (CI). The authors highlight that the concept of CI in groups emerges from the interplay of *bottom-up* and *top-down* processes. Bottom-up processes involve the aggregation of individual member characteristics, such as intelligence, social skills, and diverse knowledge, which contribute to enhanced group collaboration. Top-down processes include structures, norms, and routines that regulate collective behavior and can either enhance or impede coordination and collaboration within the group. The interaction and combination of these bottom-up and top-down aspects ultimately determine the level of a group's collective intelligence and performance.

In order to generate the optimal solution $\mathcal{G}^*$, CIR3 draws inspiration from C3 and CI and capitalizes on:

[5][208] *uses the term "cognitive convergence" to encompass various concepts that have been used to explain the important processes underlying successful collaboration, such as inter-subjectivity, co-construction, knowledge convergence, common ground, joint problem space, and transactive reasoning.*

1. The strengths of combining decentralized and centralized networks, where (a) the information flow in the group of decentralized writers facilitates the preservation of a wider range of $\mathcal{G}^+$, which is amplified by the multi-perspective analysis, and (b) the rate of convergence in the broader centralized network (between (a) and the curmudgeon agent) facilitates a faster memory alignment of $\mathcal{G}^+$;

2. The curmudgeon agent as a mechanism for introducing variation. Coupled with external evaluation tools (Figure 4.1 ⑤), the curmudgeon guides the writers towards a balanced collective convergence on the key concepts within the document, while maintaining diversity in the output.

Combined with the benefits of CIR3's hybrid topology, the cyclic process of reflection and refinement, between the writers and the curmudgeon (Figure 4.1 ⑥), amplifies the collective intelligence, and enables collaborative knowledge construction by sharing, discussing, and building upon each other's analysis, leading to a deeper understanding of the document. Additionally, this approach offers a solution to mitigate the disadvantages inherent in both centralized (potential loss of information) and decentralized (potential slow convergence) networks.

Once the inner-refinement cycle reaches either an agreement or the predefined maximum number of iterations, L, CIR3 is prompted to create a separate record of the latest refined QA pairs, $\mathcal{G}^- = \mathcal{G}^+{}_l$. This state is then passed to the outer-refinement cycle k, where the curmudgeon appends its feedback $\mathcal{F}\prime_k$ along $\mathcal{G}^-{}_{k-1}$ to the transactive memory, creating a central memory state that evolves with each outer-iteration of refinement:

$$\mathcal{H} = \{(\mathcal{G}^-{}_0, \mathcal{F}\prime_1), ..., (\mathcal{G}^-{}_{k-2}, \mathcal{F}\prime_{k-1}), (\mathcal{G}^*, \varnothing)\}_{k=1}^{K} \tag{4.3}$$

where $\varnothing$ denotes a satisfactory alignment between the curmudgeon and the writers, which then routes the subsequent operation to the termination phase, through the moderator, yielding the final output $\mathcal{G}^*$ and halting the generation process (Figure 4.1 ⑦).

The curmudgeon is equipped with an evaluation tool to help quantify the diversity of (a) generated questions, (b) generated answers, and (c) concatenated answers and input context. A lower diversity score in (c), combined with higher diversity scores in (a) and (b), would indicate high faithfulness and better coverage of the input context. To achieve this, *Vendi Score*[6] [211] is used as an evaluation tool for diversity, where the objective is to minimize diversity in (c), while maximizing it in (a) and (b).

At each iteration k, the curmudgeon evaluates $\mathcal{G}^-$ to determine the next action, leveraging the Vendi tool to augment its intrinsic knowledge and generate informed feedback. The result is then added to $\mathcal{H}$ (Equation 4.3), which invokes another cycle of inner-refinements among the writers.

Diversity Metric Selection

Vendi Score is a similarity-based diversity metric inspired by quantum statistical mechanics. It quantifies the diversity of a sample set by analyzing the eigenvalues of a similarity matrix, which captures the correlations and relationships among all data points. This approach enables Vendi Score to measure complex, multidimensional diversity that pairwise metrics may miss. Unlike many traditional diversity metrics, it does not require prevalence information on items, which makes it particularly useful when such data are unavailable or irrelevant. Vendi Score captures both the number and balance of distinct outputs, is less sensitive to sequence length variations, and operates independently of specific embedding model architectures. By evaluating the joint diversity across all attributes simultaneously, Vendi Score provides a comprehensive measure of a set's overall variability, going beyond simple pairwise comparisons to assess the full spectrum of diversity within the samples. As with any embedding-based estimator, Vendi's values depend on the encoder and its induced geometry; we therefore use a consistent encoder during evaluation and probe sensitivity where appropriate (see Chapter 5 for encoder-robustness analyses).

Convergence Properties and Limitations

CIR3 operates as a heuristic iterative framework without theoretical global optimality guarantees. However, several design safeguards mitigate common convergence issues: bounded iterations prevent infinite loops, the hybrid topology preserves diversity while enabling coordination, and the external variation signal (curmudgeon + Vendi tool) prevents premature consensus. Empirically, ablation studies (Section 4.4.4.1) demonstrate that removing these safeguards leads to

[6] SimCSE models from Princeton and BGE models are employed as foundational encoders for the Vendi score. The implementation extends this setup to include various embedders, including AMLM variants (Chapter 3). Empirically, it produces scores in the range of 1 to 2, with 1 indicating perfect similarity, typically observed between a given context and its corresponding concatenated answers.

either premature convergence with information loss or slow convergence with reduced faithfulness, validating the design choices.

Domain Robustness

CIR3's architecture is intentionally domain-agnostic to ensure cross-domain generalizability through three key mechanisms:

1. **Subtopic identification** uses LLM semantic understanding without domain-specific ontologies, reducing dependence on specialized vocabularies. A comprehensive evaluation of CIR3's subtopic identification task is presented in Chapter 5.
2. **Perspective-assigned writers** leverage general reasoning capabilities based on textual evidence rather than domain-specific knowledge bases or specialized ontologies.
3. **Curmudgeon evaluation** integrates two complementary mechanisms: intrinsic reasoning capabilities for the coverage assessment and Vendi score-based diversity optimization for comprehensive exploration.

This integrated approach ensures both comprehensive document coverage and diverse solution exploration across domains. The semantic subtopic decomposition enables flexible problem partitioning, perspective-based writing generates diverse viewpoints through general reasoning, and the dual curmudgeon mechanism balances coverage optimization with diversity preservation to prevent convergence to suboptimal local solutions.

4.3 Experiments

This section presents an empirical evaluation of CIR3's performance. The discussion opens with a description of the datasets employed, followed by an overview of the baselines selected for comparison. The subsequent subsection outlines the implementation details of CIR3. The final part introduces the evaluation metrics, which encompass statistical, encoder-based, and LLM-based methods.

4.3.1 Datasets

While widely used QA datasets like MS MARCO [212] and Natural Questions [213] offer valuable resources, they fall short for the purposes of this work due to the lack of both in-domain and specialized QA datasets, as well as insufficient coverage of comprehensive QA pairs per document. As a result, experiments are conducted exclusively on passages from four specialized datasets in Finance and Medical fields:

(1) FiQA [214]. This dataset[7] was used in the Financial Opinion Mining and Question Answering challenge at the 2018 International World Wide Web Conference. FiQA comprises 6,648 questions and 57,640 answer passages. It was curated from financial posts on platforms such as Stackexchange[8], Reddit[8], and StockTwits[8] between 2009 and 2017, with the objective of developing QA systems that can address financial queries by leveraging information from various sources such as micro-blogs, reports, and news articles;

(2) InsuranceQA [215] (InsurQA). This corpus[9] was sourced from the Insurance Library[10] website, consists of 16,889 real-world user questions and 27,413 corresponding answers written by professionals with extensive domain knowledge in the insurance industry.

(3) MedQA [216] is a free form multilingual multiple choice QA dataset[11] specifically curated for medical problem-solving, sourced from professional medical board exams. It encompasses over 61K questions, making it a valuable resource for evaluating and training models on clinical reasoning and medical knowledge in diverse contexts. The dataset is complemented by a large-scale corpus extracted from medical textbooks, supporting comprehensive reading comprehension and open-domain medical QA research.

(4) MedMCQA [217] is a large-scale, multiple-choice QA dataset[12] designed to mirror real-world medical entrance exams, notably AIIMS and NEET PG. It contains over 194K high-quality MCQs spanning 2,400 healthcare topics and 21 medical subjects, with each question accompanied by detailed explanations. The dataset is notable for its topical diversity and complexity, requiring models to demonstrate advanced reasoning abilities across a broad spectrum of medical disciplines, making it a robust benchmark for open-domain medical QA systems.

[7] https://huggingface.co/datasets/BeIR/fiqa
[8] https://stackexchange.com; https://stocktwits.com; https://reddit.com
[9] https://github.com/shuzi/insuranceQA
[10] https://www.insurancelibrary.com/
[11] https://huggingface.co/datasets/bigbio/med_qa
[12] https://huggingface.co/datasets/openlifescienceai/medmcqa

For each dataset, a subset of 2000 passages is randomly chosen for the main experiments.

4.3.2 Baselines

Prior research in this area has used varied experimental setups and has not focused on generating comprehensive sets of QA pairs from individual documents. As a result, direct comparisons between these works are challenging. Therefore, the following baselines are established for this study:

- **LLM-DP:** This baseline directly prompts META-LLAMA-3-70B-INST [151] to generate QAs without explicit reasoning or tool utilization. It serves as a measure of the LLM's ground performance.
- **qGen-aGen:** In this pipeline, QUERY-GEN-MS- MARCO-T5-LARGE-V1 from the Benchmarking IR BEIR [142] is employed to generate questions, which are then fed into META-LLAMA-3-70B to produce corresponding answers. This baseline assesses the LLM's performance when guided by an external query generation model.

4.3.3 CIR3 Implementation

CIR3 is implemented using the LangGraph [218] library, supporting both heterogeneous (CIR3-Het) and homogeneous (CIR3-Hom) configurations.

- **CIR3-Hom** uses the INSTRUCT variants of Meta-Llama-3-{70B,8B} for their strong performance and moderate resource requirements.
- **CIR3-Het**[13] leverages GPT-4o-mini [40], Gemma-3-27B-it [38], Meta-Llama-3-{70B,8B} [151], and Claude Sonnet 4 [39].

Inference is conducted with a temperature of 0.1 and nucleus sampling of 0.5. The Groq [220] API is used for Llama models and Gemma is self-hosted via vLLM [221, 222], both offering seamless integration. Generation is limited to 10 QA pairs per context, with refinement iterations set to $K = 6$ and $L = 12$.

[13] The implementation is designed for flexibility and scalability, enabling compatibility with a wide range of LLMs through external configuration alone, without requiring code modifications, as detailed in the repository [219].

4.3.4 Evaluation Metrics

This section delineates the metrics and evaluation framework used to assess CIR3's performance. The primary evaluation is first discussed, followed by common generation errors. A comprehensive evaluation of the classifier agent through cross-model semantic agreement is presented in Chapter 5.

4.3.4.1 Main Evaluation

Automatic evaluation of generated text remains a challenge as traditional metrics fail to align with human assessments. To address this limitation and provide a more comprehensive and refined evaluation of CIR3, standard metrics are augmented with LLM-based scores tailored to the specific use case.

Statistical Scorers. ROUGE-L [223], METEOR [224], and Jaccard Index [225] are used to calculate the scores between (1) the generated questions $\mathcal{Q}$ and the context c as reference, (2) the generated answers $\mathcal{A}$ and c, and (3) $\mathcal{Q}$ and $\mathcal{A}$. The mean score is then calculated over (1), (2) and (3), before calculating the average scores over each evaluation dataset.

Encoder-based Scorers. Beyond token overlap, embedding-based similarity metrics are also employed, such as BERTScore [163] and BAAI/BGE-LARGE-EN-V1.5 (denoted with BGE score in this study). The mean semantic scores are measured between (1) c and $\mathcal{Q}$, (2) c and $\mathcal{A}$, and (3) $\mathcal{Q}$ and $\mathcal{A}$. To assess the quality of QAs when considered collectively, BGE scores are also included between (4) the concatenated questions $\mathcal{Q}_\oplus = \oplus_{i=1}^{N} q_i$ and c, (5) the concatenated answers $\mathcal{A}_\oplus = \oplus_{i=1}^{N} a_i$ and c, and (6) $\mathcal{Q}_\oplus$ and $\mathcal{A}_\oplus$.

LLM-based Scorers. To further quantify the comprehensiveness and faithfulness of the generated QA pairs, the G-EVAL [148] framework is adapted by merging the task definition and evaluation criteria prompt with a Chain-of-Thoughts (CoT) prompt [101] to specify detailed evaluation steps. This modification provides greater control over the assessment process compared to the original G-EVAL, where the LLM generates the CoT automatically. The comprehensiveness of $\mathcal{G}^*$ is evaluated based on *coverage*, *depth*, *accuracy* and *coherence*. Similarly, faithfulness is evaluated based on *accuracy*, *exaggeration*, *consistency*, *justification*, *plausibility*, and *misrepresentation*. Additionally, the G-EVAL scoring function is retained, which normalizes scores using a weighted sum of token probabilities in LLM output. `GPT-4` is used with the temperature set to 0 to ensure reproducibility.

Further details on the metrics and scoring calculations used in this study are provided in B.1: Automatic Metrics. Sample prompts designed for evaluating CIR3 can be found in B.4: Evaluation Prompts.

4.3.4.2 Evaluation of Common Generation Errors

To further assess the robustness of the framework, CIR3 is evaluated in its ability to mitigate common generation errors: *hallucination*, *irrelevance*, *duplication*, and *over-specificity*. Experiments are conducted using 400 samples, with 100 samples from each of the four datasets. The generated QA pair sets are evaluated using `gpt-4o` as a model-based evaluator, employing G-Eval with detailed evaluation steps for each error type. To quantify duplication, the semantic similarity across all possible pairs of generated questions for a given document is assessed, and the averaged score is reported. The performance of the approach, **CIR3-Hom** and **CIR3-Het**, is compared against the two baselines: **LLM-DP** and **qGen-aGen**.

4.4 Results and Observations

This section presents the experimental findings, covering key results, human evaluations, and ablation studies to assess the effect of multi-perspective reasoning and the impact of introducing variation.

4.4.1 Main Results

In all tables, the best-performing model is highlighted in **bold**, with the second-best underlined.

As shown in Table 4.1, the proposed approaches outperform both baselines across all lexical metrics on both datasets. Specifically, CIR3-Hom achieves relative improvements of 6.43% on METEOR, 2.66% on ROUGE-L, and 9.62% on the Jaccard Index over the next best-performing model. Furthermore, CIR3-Het demonstrates even more substantial gains, with relative improvements of 10.41%, 3.70%, and 12.04% on the same metrics, respectively. Although the

TABLE 4.1: Evaluation results using standard metrics. † denotes significant differences ($p < 0.05$) from a paired t-test between **CIR3** and the best baseline LLM-DP.

Dataset	Model	METEOR $s(c,\mathcal{Q})$	$s(c,\mathcal{A})$	$s(\mathcal{Q},\mathcal{A})$	Avg.	ROUGE-L (F1 Scores) $s(c,\mathcal{Q})$	$s(c,\mathcal{A})$	$s(\mathcal{Q},\mathcal{A})$	Avg.	Jaccard Index $s(c,\mathcal{Q})$	$s(c,\mathcal{A})$	$s(\mathcal{Q},\mathcal{A})$	Avg.
FiQA	LLM-DP	0.1571	0.3068	0.2119	0.2252	0.1951	0.3189	0.2781	0.2640	0.4377	0.5286	0.4881	0.4847
	qGen-aGen	0.1288	0.3383	0.1613	0.2094	0.1771	0.4003	0.2690	0.2821	0.4161	0.5391	0.4703	0.4751
	CIR3-Hom	0.1935	0.3791	0.2767	**0.2831†**	0.2153	0.3771	0.2893	**0.2939**	0.5511	0.6112	0.5983	**0.5868†**
	CIR3-Het	0.2140	0.3905	0.2938	**0.2994†**	0.2190	0.3913	0.2952	**0.3018**	0.5633	0.6959	0.5983	**0.6191†**
InsurQA	LLM-DP	0.2422	0.3972	0.2717	0.3037	0.2877	0.4984	0.3447	0.3769	0.4784	0.5920	0.4987	0.5230
	qGen-aGen	0.1433	0.3134	0.1283	0.1949	0.1898	0.4903	0.2463	0.3088	0.3885	0.5749	0.4729	0.4787
	CIR3-Hom	0.3197	0.4391	0.3632	**0.3739†**	0.2950	0.4891	0.3972	**0.3937**	0.5261	0.6716	0.6104	**0.6027†**
	CIR3-Het	0.3278	0.4579	0.3802	**0.3887†**	0.3143	0.5002	0.4031	**0.4059**	0.5354	0.6769	0.6210	**0.6111†**
MedQA MedMCQA	LLM-DP	0.1506	0.3042	0.2077	0.2208	0.1959	0.3161	0.2715	0.2611	0.4351	0.5303	0.4814	0.4823
	qGen-aGen	0.1231	0.3353	0.1575	0.2053	0.1708	0.2982	0.2708	0.2466	0.4136	0.5390	0.4680	0.4735
	CIR3-Hom	0.1958	0.3847	0.2765	**0.2857†**	0.2204	0.3739	0.2887	**0.2943**	0.5512	0.6146	0.6019	**0.5892†**
	CIR3-Het	0.2070	0.3934	0.2766	**0.3739†**	0.2317	0.3798	0.3050	**0.3055**	0.5582	0.6220	0.6104	**0.6212†**

observed overlap might not suggest a high degree of similarity, it is important to consider the limitations of lexical metrics, which are inherently less effective when evaluating generative tasks.

TABLE 4.2: Evaluation results using embedding-based metrics. † denotes significant differences ($p < 0.05$) from a paired t-test between **CIR3** and the best baseline LLM-DP.

Dataset	Model	BERTScore (F1 Scores) $s(c,\mathcal{Q})$	$s(c,\mathcal{A})$	$s(\mathcal{Q},\mathcal{A})$	Avg.	BGE Semantic Similarity $s(c,\mathcal{Q})$	$s(c,\mathcal{A})$	$s(\mathcal{Q},\mathcal{A})$	Avg.	$s(c,\mathcal{Q}_\oplus)$	$s(c,\mathcal{A}_\oplus)$	$s(\mathcal{Q}_\oplus,\mathcal{A}_\oplus)$	Avg.
FiQA	LLM-DP	0.8415	0.8597	0.8701	0.8571	0.6858	0.6847	0.7872	0.7192	0.7548	0.8078	0.8488	0.8038
	qGen-aGen	0.8339	0.8617	0.8472	0.8475	0.6932	0.7051	0.7358	0.7113	0.7462	0.8087	0.8183	0.7910
	CIR3-Hom	0.8702	0.9171	0.9088	**0.8987†**	0.8312	0.8542	0.8115	**0.8323**	0.8291	0.9118	0.9264	**0.8891†**
	CIR3-Het	0.9085	0.9412	0.9378	**0.9292†**	0.8555	0.8689	0.8407	**0.8551**	0.8554	0.9384	0.9587	**0.9175†**
InsurQA	LLM-DP	0.8511	0.8810	0.8779	0.8700	0.7388	0.7540	0.8097	0.7675	0.8173	0.8948	0.8675	0.8598
	qGen-aGen	0.8282	0.8757	0.8472	0.8503	0.7231	0.7404	0.7344	0.7326	0.7708	0.8539	0.7487	0.7911
	CIR3-Hom	0.8972	0.9298	0.9175	**0.9148†**	0.7591	0.7736	0.8616	**0.7980**	0.8450	0.9395	0.9072	**0.8972†**
	CIR3-Het	0.9218	0.9414	0.9352	**0.9328†**	0.7833	0.8008	0.8809	**0.8217**	0.8389	0.9530	0.9215	**0.9044†**
MedQA MedMCQA	LLM-DP	0.8358	0.8525	0.8696	0.8526	0.6850	0.6795	0.7848	0.7165	0.7486	0.8039	0.8483	0.8003
	qGen-aGen	0.7845	0.8633	0.8347	0.8275	0.6528	0.6734	0.6869	0.6710	0.7288	0.7540	0.7770	0.7533
	CIR3-Hom	0.8776	0.9256	0.9235	**0.9089†**	0.7777	0.7592	0.8511	**0.7960**	0.8559	0.9292	0.8851	**0.8901†**
	CIR3-Het	0.8971	0.9454	0.9525	**0.9316†**	0.8053	0.7835	0.8708	**0.8199**	0.8787	0.9552	0.9056	**0.9132†**

Further analysis in Table 4.2 shows that CIR3 consistently surpasses other models in semantic similarity metrics. CIR3 achieves an average improvement of 5.94% on BERTScore and 8.33% on BGE compared to the second-best model. This trend extends to contextual semantic similarity between the context and concatenated answers, suggesting that CIR3's generated answers are more faithful to the input text, potentially indicating lower hallucination and improved comprehensiveness.

Table 4.3 (LLM-based evaluation results) further supports these findings, showing that CIR3-Hom achieves average improvements of 21.85% in comprehensiveness and 16.62% in faithfulness, while CIR3-Het attains 23.33% and 17% improvements, respectively, outperforming the second-best model, LLM-DP. These results provide additional validation for Method 4.2.3 and Equation 4.1, wherein the curmudgeon, utilizing a diversity-based evaluation tool, directs the generation of diverse QAs (Comprehensiveness) while ensuring the alignment of the answers with the context (Faithfulness).

TABLE 4.3: LLM-based evaluation results for *comprehensiveness* and *faithfulness*.

Dataset	Model	Comprehensive	Faithful	Avg.
FiQA	LLM-DP	0.7169	0.8030	0.7599
	qGen-aGen	0.5290	0.8414	0.6852
	CIR3-Hom	0.9312	0.9762	**0.9537**
	CIR3-Het	0.9419	0.9749	**0.9584**
InsurQA	LLM-DP	0.7317	0.8175	0.7746
	qGen-aGen	0.5560	0.8763	0.7161
	CIR3-Hom	0.9389	0.9879	**0.9634**
	CIR3-Het	0.9501	0.9893	**0.9697**
MedQA MedMCQA	LLM-DP	0.6807	0.7961	0.7384
	qGen-aGen	0.5052	0.8371	0.6711
	CIR3-Hom	0.9148	0.9511	**0.9329**
	CIR3-Het	0.9372	0.9629	**0.9500**

Interestingly, LLM-DP demonstrates superior performance compared to qGen-aGen in all tests. This implies that the added query generator may not be beneficial, possibly due to the limitations of the T5 [21] model in uncovering deeper key concepts in both domains.

Analysis also reveals, in Tables 4.1 and 4.2, that CIR3's questions are significantly more aligned with the context compared to both baselines. This indicates that CIR3's deep engagement with the input document helps bridge the gaps in machine reading comprehension, which results in more comprehensive and relevant question generation.

The results presented in Tables 4.1, 4.2, 4.3, provide evidence of the effectiveness of CIR3. These results further demonstrate that CIR3-Het consistently outperforms CIR3-Hom, a performance gap attributable to the advantages conferred by architectural and behavioral diversity among heterogeneous agents, where differing model architectures, training data, and inductive biases result in varied reasoning strategies and error profiles, enabling the system to explore a broader solution space and mitigate redundancy or shared failure modes common in homogeneous configurations. This diversity enhances robustness through inter-agent disagreement, encourages complementary specialization, and supports more effective ensemble decision-making. Prior work highlights how heterogeneity improves collaborative reasoning, debate, and problem-solving performance in LLM-based agents [226]. Such findings also align with our hypotheses in **H.2** and **H.3**, where cognitive diversity often correlates with improved group performance.

4.4.2 Common Generation Error Analysis

The results presented in Table 4.4 demonstrate that both CIR3 variants significantly outperform the baseline methods across all error categories and domains. CIR3-Het achieves the highest performance, with scores exceeding 0.94 across all metrics for both Finance and Medical domains.

Table 4.4: LLM-based evaluation results for common generation errors (*semantic duplication*, *hallucinated answers*, *irrelevant QAs*, *over-specific* and *over-generalized answers*). Higher scores indicate better performance.

Dataset	Model	Duplication	Hallucination	Irrelevance	Over Specificity
FiQA InsurQA	LLM-DP	0.7846	0.8088	0.8006	0.8976
	qGen-aGen	0.4771	0.8519	0.8481	**0.9781**
	CIR3-Hom	**0.9515**	**0.9796**	**0.9783**	0.9549
	CIR3-Het	**0.9689**	**0.9853**	**0.9825**	0.9533
MedQA MedMCQA	LLM-DP	0.7012	0.7935	0.7933	0.8860
	qGen-aGen	0.4919	0.8466	0.8452	**0.9703**
	CIR3-Hom	**0.9317**	**0.9690**	**0.9630**	0.9561
	CIR3-Het	**0.9481**	**0.9811**	**0.9847**	0.9598

Duplication Analysis. The most substantial performance gap appears in duplication scores, where CIR3 achieves scores between 0.93 and 0.97, while LLM-DP scores range from 0.70-0.79, and qGen-aGen performs poorly with scores below 0.5. This suggests that T5-based question generation tends to produce highly similar questions, limiting diversity in the generated QAs.

Over-specificity. Interestingly, qGen-aGen scores high in over-specificity (~ 0.97). This can be attributed to the characteristics of the fine-tuned T5 question generation model (query-gen-ms-marco-t5-large-v1), which tends to produce more generic, template-driven questions. Although this approach helps prevent overfitting to specific contextual details, it comes at the steep cost of diversity, as evidenced by the low duplication scores.

Hallucination and Irrelevance Control. CIR3 variants exhibit superior hallucination prevention compared to baseline methods, with CIR3-Het achieving scores of 0.98+, substantially higher than LLM-DP (~ 0.80) and baseline qGen-aGen (~ 0.84).

The results demonstrate that CIR3's approach to QAG effectively balances all evaluation criteria, producing high-quality, diverse, and contextually appropriate QAs while minimizing common generation errors.

4.4.3 Human Evaluation

Human evaluation is further conducted on 80 samples from the InsurQA corpus and the corresponding generated QA pairs by CIR3 and LLM-DP. Eight experts in finance[14] assess 10 sets of QA pairs each, focusing on comprehensiveness and faithfulness. Comprehensiveness is evaluated based on three aspects: *coverage*, *depth*, and *coherence*. Similarly, faithfulness is assessed

[14] Volunteers have 2 to 6 years of experience in the finance domain, all based in Europe

based on: *accuracy*, *representation*, and *diversification*. Each aspect is scored on a scale from 1 (worst) to 5 (best). A partial excerpt of the evaluation guidelines is given in B.3: Human Evaluation Guidelines.

TABLE 4.5: Human evaluation results on 80 sets of QA pairs generated by CIR3 and LLM-DP. The ratings (1 to 5) are normalized between 0 and 1. The scores are analyzed using a paired t-test (p-values are presented).

	Aspect	LLM-DP	CIR3	p-value
Comprehensiveness	Coverage	0.7875	0.9375	0.0033
	Depth	0.7750	0.9125	0.0038
	Coherence	0.7625	0.9250	0.0023
	Avg.	0.7750	**0.9250**	
Faithfulness	Accuracy	0.7500	0.9125	0.0020
	Representation	0.7875	0.9125	0.0042
	Diversification	0.8250	0.8875	0.0104
	Avg.	0.7875	**0.9041**	

Table 4.5 shows the average scores and paired t-test results, aligning with the findings in Table 4.3. CIR3 demonstrates significant improvement over the baseline LLM-DP, with an increase of 15% on comprehensiveness and 11.66% on faithfulness.

4.4.4 Ablation Studies

4.4.4.1 Ablation Studies: Perspective and External Variation Effect

To provide additional support for the hypotheses in **H.2** and **H.3**, an ablation study is conducted with two variations of CIR3:

1. **CIR3 w/o perspectives**. Following [203], in this variation, the impact of multi-perspective reasoning is assessed. The moderator's prompt is modified by removing the section that assigns diverse perspectives to the writer agents. To ensure a fair comparison, the same number of writers as in the original model (determined by the number of identified subtopics) is maintained;

2. **CIR3 w/o Curmudgeon**. In this variation, the curmudgeon agent is disabled to evaluate the effect of introducing external variation to the writer's sub-network.

For this study, 200 samples are randomly selected, equally split between both datasets, with refinement cycles between writers capped at 12 for each input.

TABLE 4.6: Effect of multi-perspective reasoning and Curmudgeon on *Comprehensiveness* and *Faithfulness*.

Model	Comprehensiveness	Faithfulness	Avg.
LLM-DP	0.7399	0.8221	0.7810
CIR3	**0.9451**	**0.9895**	**0.9673**
CIR3 w/o perspectives	0.9115	0.9653	0.9384
CIR3 w/o Curmudgeon	0.8370	0.9046	0.8708

The results in Table 4.6 demonstrate that CIR3 surpasses the two alternative variations. Nonetheless, both variations outperform the baseline LLM-DP, providing some support for the hypotheses.

Effect of multi-perspective reasoning. Table 4.6 shows that *CIR3 w/o perspectives* yields inferior results compared to CIR3, suggesting that multi-perspective group debate contributes to a comprehensive and faithful output, as proposed in **H.2**.

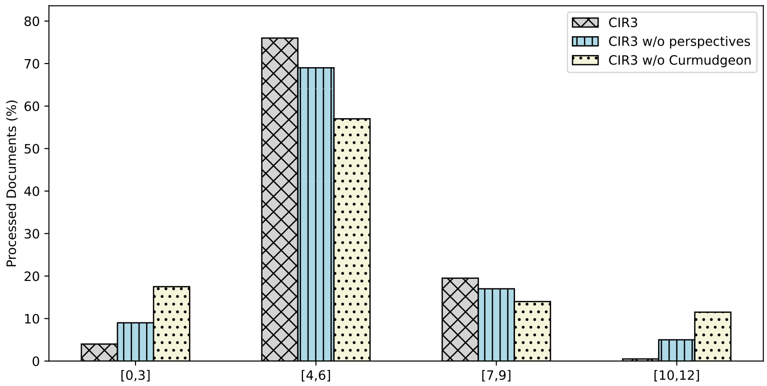

FIGURE 4.2: Number of inner-refinement cycles (x-axis), given as intervals, required to process the input documents (y-axis), given as percentage.

TABLE 4.7: Effect of Multi-Perspective Reasoning and Curmudgeon on Document Distribution per Cycle.

	Context Distribution Per Refinement Cycle Ranges (%)			
Model	$[0, 3]$	$[4, 6]$	$[7, 9]$	$[10, 12]$
LLM-DP	100	-	-	-
CIR3	04.00	76.00	19.50	00.50
CIR3 w/o perspectives	09.00	69.00	17.00	05.00
CIR3 w/o Curmudgeon	17.50	57.00	14.00	11.50

Effect of variation. Removing the disruptive signal, in *CIR3 w/o Curmudgeon*, significantly impairs performance, reducing faithfulness by 8.49% and comprehensiveness by 10.81%. This can be explained by examining the number of refinement cycles (given as intervals) required to process the input documents, as in Figure 4.2. Compared to CIR3, and *CIR3 w/o perspectives*, *CIR3 w/o Curmudgeon* shows a significant increase in the number of contexts falling within the refinement cycle ranges $[0, 3]$ and $[10, 12]$, as shown in Table 4.7. For the interval $[0, 3]$, *CIR3* *w/o Curmudgeon* exhibits a 13.5% increase compared to CIR3 and an 8.5% increase compared to *CIR3 w/o perspectives*. Similarly, for the interval $[10, 12]$, *CIR3 w/o Curmudgeon* shows an 11% increase over CIR3 and a 6.5% increase over *CIR3 w/o perspectives*. This aligns with **H.3**, where the absence of variation can result in either (1) a potential immature collective convergence (collapse) and loss of information, characterized by a small number of iterations and potentially low comprehensiveness scores, or (2) a potential slow convergence, characterized by a large number of iterations and a high likelihood of low faithfulness.

4.4.4.2 Ablation Studies: Curmudgeon Strategies

To evaluate the individual contributions of the curmudgeon agent and the Vendi diversity tool in CIR3, comprehensive ablation studies are conducted under four experimental conditions using 200 documents uniformly sampled from the four datasets (50 each):

1. **CIR3 (baseline)**: The curmudgeon agent combines its intrinsic knowledge with the Vendi diversity tool.
2. **CIR3 Intrinsic Only**: The curmudgeon agent operates solely using its intrinsic knowledge without access to the Vendi diversity tool.
3. **CIR3 Vendi Only**: A simplified configuration using only the Vendi tool for diversity measurement (binary feedback[15]), with no curmudgeon agent providing qualitative feedback.
4. **CIR3 Random Rejection**: A control condition employing random feedback with an acceptance probability of 0.35, eliminating both agent reasoning and diversity measurement.

All configurations utilizing LLM agents employed GPT-4o-mini for its superior reasoning capabilities and cost-efficiency. Ablation studies reveal distinct behavioral patterns across the four configurations, measured by document distribution across outer refinement iteration intervals and corresponding comprehensiveness scores.

Comprehensiveness Performance. As shown in Table 4.8, the *CIR3* baseline achieved the highest average comprehensiveness score at 0.9528, demonstrating the effectiveness of combining

[15] "*QA pair meets or not diversity and / or alignment criteria.*"

TABLE 4.8: Ablation study of Curmudgeon feedback strategies.

Strategy	Avg. Comp ↑	Avg. Faith↑	Avg. cycles ↓	% docs processed in:		
				[1,2]	[3,4]	[5,6]
CIR3	0.9528	0.9895	2.35	4.0	57.0	39.0
CIR3 Intrinsic Only	0.9223	0.9696	2.45	5.5	44.5	50.0
CIR3 Vendi Only	0.8893	0.9331	2.85	0.5	14.5	85.0
CIR3 Random Rejection	0.8167	0.8969	1.37	73.0	17.0	10.0

the reasoning of the curmudgeon agent with the diversity measurement tool. *CIR3 Intrinsic Only* performed moderately well (0.9223) but showed reduced quality without diversity guidance. *CIR3 Vendi Only* achieved lower comprehensiveness (0.8893), suggesting that diversity measurement alone is insufficient for an effective quality assessment. The *random rejection* baseline predictably performed the lowest (0.8167), reinforcing the need for intelligent feedback mechanisms.

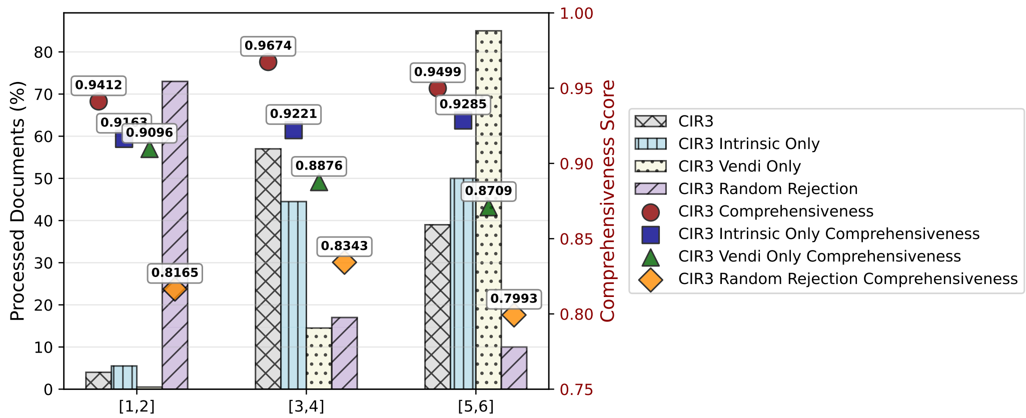

FIGURE 4.3: Comparative analysis of four CIR3 variants, distinguished by their Curmudgeon strategies, across defined outer-refinement cycle ranges. Bars show the percentage of documents processed within each cycle range (left y-axis), and scatter points denote the average comprehensiveness score (right y-axis).

Document Distribution Patterns. As illustrated in Figure 4.3, the distribution of documents across refinement intervals [1,2], [3,4], and [5,6] revealed distinct processing patterns. Random rejection processed 73% of documents within the first two cycles, indicating premature convergence (Comprehensiveness 0.8165). In contrast, *CIR3 Vendi Only* required extensive refinement, with 85% of the documents needing 5-6 cycles, indicating slow convergence (Comprehensiveness 0.8709). The original *CIR3* demonstrated a more balanced convergence, with document distributions of 4%, 57% and 39% across the [1,2], [3,4], and [5,6] intervals, achieving respective comprehensiveness scores of 0.9412, 0.9674, and 0.9499.

TABLE 4.9: Document count percentage and comprehensiveness score progression for documents requiring exactly 5 outer refinement iterations.

Model Variant	Doc %	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Improvement
CIR3	24.5%	0.7801	0.8795	0.9391	0.9437	0.9573	**+0.1772**
CIR3 Intrinsic Only	29.0%	0.7819	0.8553	0.9180	0.9193	0.9291	+0.1472
CIR3 Vendi Only	32.0%	0.7923	0.8333	0.8554	0.8622	0.8770	+0.0847
CIR3 Random Rejection	7.0%	0.7811	0.7886	0.7909	0.7991	0.7975	+0.0164

Complex Document Analysis. For the most challenging documents[16], which required exactly five refinement iterations ($K - 1$), *CIR3* again showed superior performance, with a significant improvement from 0.7801 to 0.9573 (+17.72% - Table 4.9). In contrast, *CIR3 Vendi Only* exhibited the smallest improvement (0.7923 $\rightarrow$ 0.8770), as detailed in Figure 4.4. This highlights the critical role of intelligent agent feedback in difficult refinement tasks.

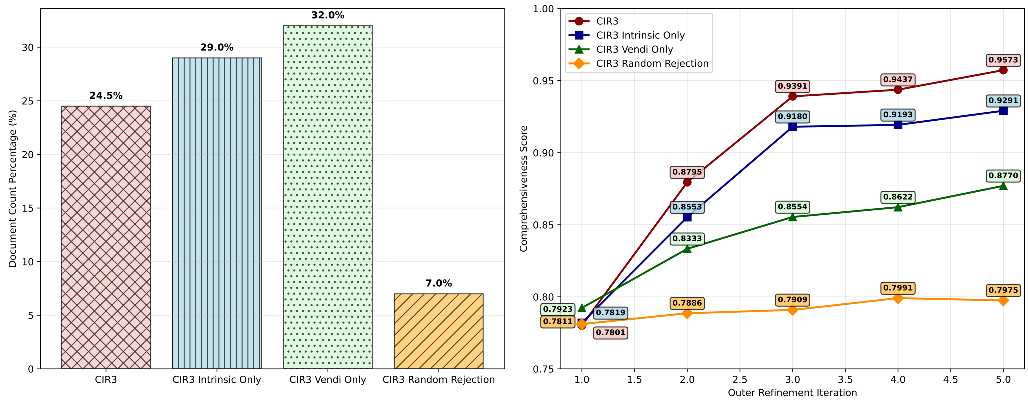

FIGURE 4.4: To analyze the behavior of different Curmudgeon strategies, this figure breaks down performance on documents requiring exactly five outer refinement iterations, showing the percentage of such documents (left) and the corresponding evolution of their comprehensiveness score (right).

Key Finding. The ablation results demonstrate that: (1) the intrinsic reasoning capabilities of the curmudgeon agent are the main driver of quality improvements, (2) the Vendi diversity tool provides measurable enhancement when combined with the reasoning of the agent, (3) neither component alone achieves the performance of the integrated system, and (4) the substantial performance gap between random rejection and all other conditions validates the importance of intelligent feedback in iterative refinement processes.

These findings confirm the synergistic benefit of integrating both components in CIR3. This combination outperforms either element in isolation while achieving balanced convergence, which is reflected in the high scores.

[16]These were among the most challenging documents, processed in five iterations, just below the maximum limit ($K = 6$) used to halt refinement.

4.5 Conclusion and Future Work

This work presented CIR3, a novel system for comprehensive and faithful QAG from information dense documents. A key contribution lies in addressing the more challenging QAG task compared to traditional QG, effectively navigating a constrained search space for unique and relevant QA pairs. Notably, to the best of our knowledge, CIR3 is the first proposed QAG approach employing multi-agent LLMs, orchestrating information flow via transactive reasoning, multi-perspective assessment, and balanced collective convergence. This chapter demonstrates that integrating an external signal significantly enhances convergence and diversity within the agent group, enabling efficient agreement on comprehensive and faithful QA pairs representing core text concepts, a crucial aspect of CIR3's design. To improve alignment with human evaluation, a custom metric was developed leveraging encoder and LLM-based scores on individual and concatenated QA pairs, providing a refined quality assessment.

Empirical results confirm CIR3's significant performance gains over strong baselines. A comprehensive evaluation of the classifier agent, covering cross-model semantic agreement, cross-domain generalizability, and sensitivity to embedding choice using general-purpose and domain-informed (AMLM-Fusion; Chapter 3) encoders, is provided in Chapter 5.

Future research aims to broaden the applicability of CIR3 across a diverse range of tasks, such as summarization, information retrieval, and multi-modal applications.

Chapter 5

Consensus without Gold: Semantic Agreement and Reliability for Language Model Evaluation

Abstract. *Evaluation in specialized NLP domains is constrained by the scarcity of high-quality human annotations, which limits the measurement and comparison of semantic systems. We propose a consensus-based, gold-free evaluation framework that induces a semantic baseline from multiple model outputs and quantifies its stability using inter-rater reliability. The approach represents candidate outputs with dense embeddings, forms multi-granularity clusters to derive consensus concepts, and computes reliability with Krippendorff's Alpha on the induced labels. Target systems are compared to the consensus using complementary agreement metrics that capture both many-to-many overlap and one-to-one alignment, with scores averaged across clustering thresholds for robustness. We validate the framework through two complementary studies using subtopic identification as the application domain. Study A assesses cross-domain behavior, while Study B examines sensitivity to embedding backbones by comparing general-purpose and domain-informed encoders, with both following the same evaluation procedure. Empirical findings indicate that the framework produces stable consensus baselines ($\alpha \approx 0.92 - 0.94$) and maintains consistent alignment with target systems across domains and embedding models. These outcomes support systematic, annotation-free evaluation, enabling reliable assessment of semantic fidelity. Differences across embedding backbones remain low, underscoring the framework's generalizability and robustness.*

5.1 Introduction

Evaluation is a central bottleneck for natural language processing in specialized domains. High-quality human annotation requires domain expertise, which makes large-scale gold-standard datasets expensive and slow to produce. This scarcity undermines the ability to measure progress, compare systems, and identify systematic failure modes, creating a practical obstacle for research and deployment in regulated or technical domains.

Existing alternatives reduce annotation cost but have limitations for assessing semantic fidelity. Self-consistency checks can reveal certain reasoning failures but do not directly assess whether generated outputs preserve the source meaning [12]. Reference-free metrics based on intrinsic model statistics show only modest correlation with semantic quality in many settings [13, 14]. Approaches that treat a single LLM as a judge show promise but risk model-specific bias and lack formal reliability guarantees [15, 150]. SEED [227] illustrates how continuous embeddings derived from a single pretrained encoder can support few-shot, prototype-based claim verification, yet it remains a supervised, label-dependent classifier over individual claim-evidence pairs and does not offer a gold-free or consensus-based view of semantic structure across multiple model outputs. ChatEval [132] examines voting and debate strategies among language models for discrete decisions, relying on categorical labels or exact matches. These techniques highlight the value of multi-model agreement, but they do not include formal reliability estimation and do not assess consistency across diverse embedding spaces.

This chapter proposes a general, gold-free evaluation methodology that addresses these limitations by combining multi-model aggregation with principled reliability estimation. Rather than relying on a single judge or on scarce human labels, the method induces a semantic baseline from the collective outputs of multiple diverse models and quantifies the stability of that baseline using inter-rater reliability (IRR) measures. Key design goals are: the ability to operate without human labels, mitigation of single-model bias via multiple evaluators, statistical assessment of consensus reliability, and accommodation of semantic equivalence despite lexical variation.

Methodologically, the framework represents candidate outputs with dense semantic embeddings, groups them at multiple granularities using agglomerative clustering, and converts cluster assignments into concept labels for reliability analysis. Reliability is estimated with Krippendorff's Alpha computed over induced cluster labels, providing a principled measure that handles varying annotator counts and missing assignments [228, 154]. Target systems are then compared to the induced consensus using complementary agreement metrics that capture both many-to-many and one-to-one semantic correspondence. Averaging results across clustering granularities yields stable, interpretable summary scores.

The approach is intended as a general evaluation layer applicable wherever semantic equivalence is the primary concern. In this chapter, we validate the methodology through a focused application to subtopic identification, where outputs are generative and lexical variation is common. We also report cross-domain experiments and sensitivity analyses for different embedding backbones to assess the method's robustness and generality.

5.1.1 Research Questions

To guide the study, we formulate the following research questions:

1. **RQ1:** How reliable is a semantic consensus induced from multiple LLMs for evaluation of generative semantic tasks when agglomerative clustering is applied across different threshold configurations?
2. **RQ2:** Does a holistic, consensus-based evaluation provide complementary, unique insights compared to direct pairwise comparison metrics when assessing target systems against an induced consensus baseline?
3. **RQ3:** How generalizable is the proposed framework across domains and how sensitive are conclusions to the choice of embedding backbone, including domain-informed versus general-purpose encoders?

To address these questions, we make the following contributions:

- We develop a two-stage, cross-model consensus methodology that induces a semantically grounded baseline from multiple LLM outputs using multi-granularity clustering and principled reliability estimation.
- We operationalize reliability assessment by computing a flexible measure of IRR (Krippendorff's Alpha) on induced conceptual labels and aggregating reliability estimates across clustering thresholds to improve stability.

- We propose complementary agreement metrics for evaluating target systems against the induced consensus, capturing both many-to-many semantic overlap and optimal one-to-one alignment, and average scores across thresholds for robust scoring.

- We validate the methodology empirically through cross-domain experiments and encoder-sensitivity analyses, using subtopic identification as a focused application to demonstrate the framework's behaviour in generative outputs.

The remainder of the chapter presents the proposed framework and its empirical validation. Section 5.2 presents the two-stage semantic evaluation approach, detailing both pairwise and holistic evaluations. Section 5.3 describes the experimental setup, including datasets, consensus models, and evaluation protocols. Section 5.4 reports results and analyses across domains and embedding models. Finally, Section 5.5 concludes with a discussion of findings and future research directions.

5.1.2 Objectives and Scope

The primary objective of this chapter is to introduce and validate a consensus-based evaluation framework that operates without human gold standards (addresses RQ 1 and RQ 2). We also pursue two secondary objectives that connect this chapter to the rest of the dissertation: (i) benchmark the CIR3 classifier agent (Chapter 4) within the consensus framework, and (ii) assess embedding-model sensitivity by comparing general-purpose (BGE-Large) with domain-informed (AMLM-Fusion from Chapter 3) encoders (addresses RQ 3). This design positions consensus as the central contribution while establishing coherence with prior chapters.

5.2 Methodology

This section presents a two-stage cross-model semantic evaluation framework designed to address the research questions outlined above. The framework employs two complementary evaluation approaches: (1) direct pairwise semantic agreement evaluation, which compares individual model outputs using standard similarity metrics, and (2) holistic consensus-based evaluation, which establishes a reliable consensus baseline through multi-model clustering and evaluates target systems against this consensus using semantic agreement metrics.

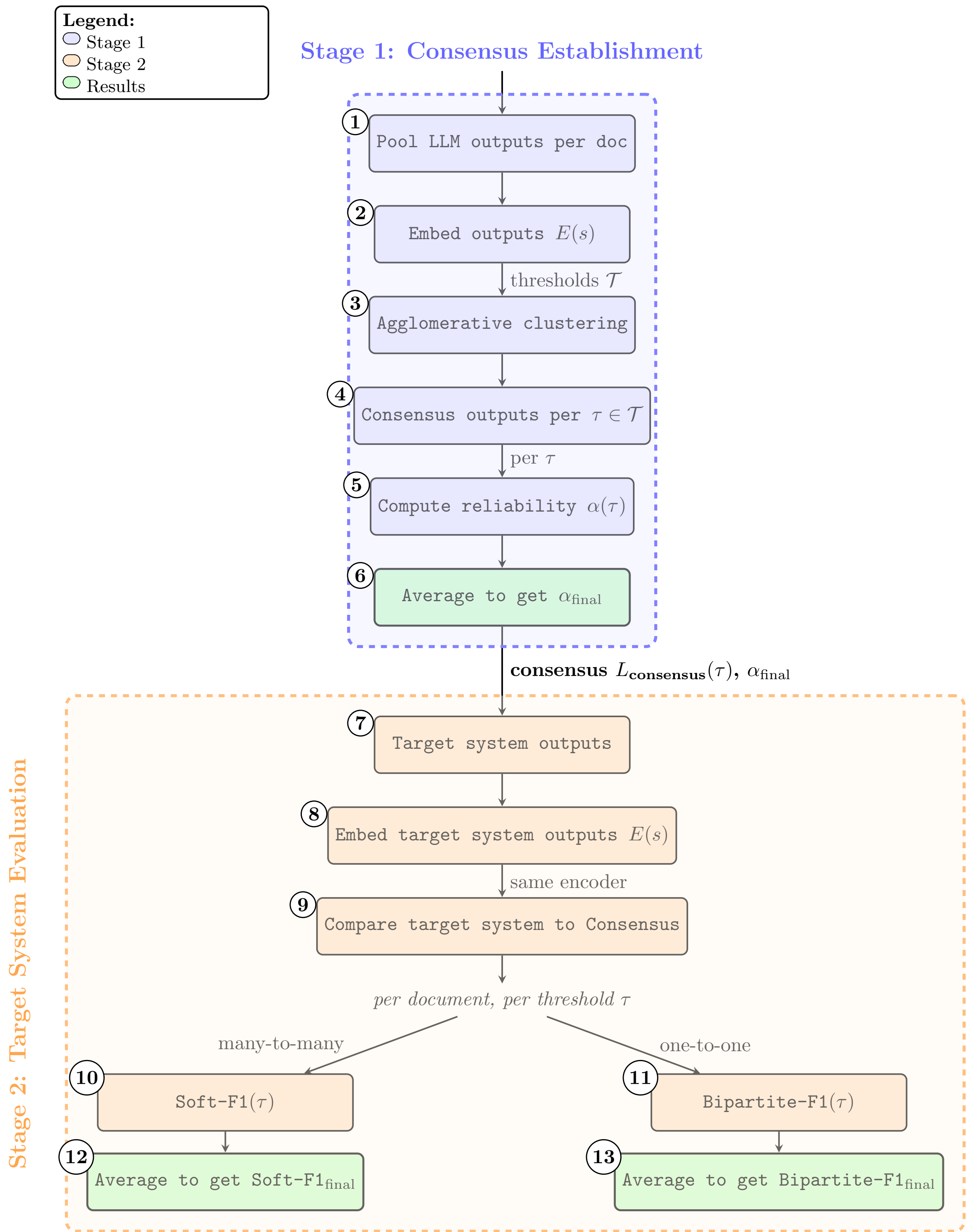

FIGURE 5.1: Two-stage evaluation framework. In Stage 1 (Consensus Establishment; steps 1–6), outputs from multiple LLMs are pooled, embedded, and clustered across a set of thresholds ($\mathcal{T}$) to derive a consensus baseline, with reliability quantified by the averaged Krippendorff's Alpha (α_{final}). In Stage 2 (Target System Evaluation; steps 7–13), the target system's outputs are compared against the consensus outputs to compute final Soft-F1 and Bipartite-F1 agreement scores, averaged across all thresholds.

Figure 5.1 provides a high-level overview of the holistic, two-stage consensus-based evaluation framework (Stage 1: consensus establishment; Stage 2: target-system evaluation). The pairwise evaluation approach is described separately in Section 5.2.2.

5.2.1 Output Representation: Semantic Embeddings

The framework is encoder-agnostic and can operate with any embedder. In our validation studies, we use a general-purpose encoder (`BGE-Large`) as the baseline and compare it with domain-informed alternatives (`AMLM-Fusion`; Chapter 3) to assess embedding sensitivity and cross-chapter integration, addressing RQ 3.

Each output s is represented as a dense vector embedding using a sentence encoder:

$$E(s) = \text{SentenceEncoder}(s) \in \mathbb{R}^d \tag{5.1}$$

where $d \in \{768, 1024\}$ is the embedding dimension. These embeddings are used throughout the evaluation framework: in Stage 1 for clustering outputs (Figure 5.1 ②) and agglomerative clustering (Figure 5.1 ③), and in Stage 2 for embedding target system outputs (Figure 5.1 ⑦) and comparing them against consensus outputs (Figure 5.1 ⑧).

The cosine similarity between embeddings is computed as:

$$\text{sim}(s_i, s_j) = \frac{E(s_i) \cdot E(s_j)}{\|E(s_i)\|_2 \|E(s_j)\|_2} \tag{5.2}$$

where $E(\cdot)$ denotes the embedding function that maps s to its vector representation. Before computing similarity, all embeddings are L2-normalized to unit length, so cosine similarity is numerically equal to the dot product between the resulting unit vectors, with values in $[-1, 1]$. To obtain a similarity score bounded between 0 and 1, we rescale the cosine similarity as:

$$\tilde{\text{sim}}(s_i, s_j) = \frac{\text{sim}(s_i, s_j) + 1}{2} \tag{5.3}$$

5.2.2 Pairwise Semantic Agreement

Direct agreement between the target system and each reference model (the individual LLMs used as evaluators) is quantified using four complementary metrics. The similarity threshold θ is used for determining valid matches in Soft-F1 and Bipartite-F1 calculations.

Given target system outputs $L_{\text{target}} = \{s_{1,i}\}_{i=1}^{m}$ and reference model outputs $L_{\text{ref}} = \{s_{2,j}\}_{j=1}^{n}$, where $m = |L_{\text{target}}|$ and $n = |L_{\text{ref}}|$ denote the number of outputs in each collection, respectively, we compute the following four metrics:

1. Jaccard Similarity. The overlap of unique outputs is measured as an exact string matching baseline, providing a lower-bound comparison for semantic similarity methods:

$$J(L_{\text{target}}, L_{\text{ref}}) = \frac{|L_{\text{target}} \cap L_{\text{ref}}|}{|L_{\text{target}} \cup L_{\text{ref}}|} \tag{5.4}$$

where $J \in [0, 1]$ with $J = 1$ indicating perfect overlap and $J = 0$ indicating no overlap.

2. Soft-F1 Score. Precision and recall are measured based on cosine similarity between output embeddings using the threshold θ. This approach allows many-to-many matching, where each output can match with multiple outputs from the other list based on semantic similarity:

$$P_{\text{soft}} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left(\max_{j} \text{sim}(s_{1,i}, s_{2,j}) \geq \theta\right) \tag{5.5}$$

$$R_{\text{soft}} = \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}\left(\max_{i} \text{sim}(s_{1,i}, s_{2,j}) \geq \theta\right) \tag{5.6}$$

$$F1_{\text{soft}} = \begin{cases} \dfrac{2 \cdot P_{\text{soft}} \cdot R_{\text{soft}}}{P_{\text{soft}} + R_{\text{soft}}} & \text{if } P_{\text{soft}} + R_{\text{soft}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

where $\theta \in [0, 1]$ denotes the cosine similarity threshold used for matching and $\mathbb{I}(\cdot)$ is the indicator function, returning 1 if the condition is true and 0 otherwise. Higher Soft-F1 scores indicate better semantic agreement, with $F1_{\text{soft}} = 1$ representing perfect many-to-many semantic alignment and $F1_{\text{soft}} = 0$ indicating no cosine similarity matches above the threshold θ.

Note that since multiple outputs can match the same counterpart, precision and recall may both appear high even when semantic overlap is partial, making Soft-F1 particularly suitable for capturing semantic flexibility and handling cases where multiple outputs express similar concepts. However, this flexibility may not reflect the true correspondence quality when strict alignment is desired. To complement this approach, we employ bipartite matching to enforce optimal one-to-one semantic correspondence, providing a more stringent evaluation that ensures each output is matched to at most one counterpart.

3. Bipartite Matching (Hungarian Algorithm-based F1). To evaluate strict one-to-one semantic alignment between outputs from two lists, we employ the Hungarian algorithm, which computes the optimal assignment in a weighted bipartite graph. Here, the two disjoint sets of nodes correspond to the outputs from the two lists:

$$S_1 = \{s_{1,1}, \ldots, s_{1,m}\}, \quad S_2 = \{s_{2,1}, \ldots, s_{2,n}\}.$$

We construct a *cost matrix* based on cosine similarity:

$$C_{ij} = 1 - \mathrm{sim}(s_{1,i}, s_{2,j}), \tag{5.8}$$

where $\mathrm{sim}(s_{1,i}, s_{2,j}) \in [0, 1]$ is the cosine similarity between output embeddings. High similarity corresponds to low cost, so minimizing total cost is equivalent to maximizing total similarity.

The optimal one-to-one assignment π^* is obtained by solving:

$$\pi^* = \arg\min_{\pi \in \Pi} \sum_{i=1}^{\min(m,n)} C_{i,\pi(i)}, \tag{5.9}$$

where Π is the set of all valid one-to-one assignments (permutations) between the two lists. This ensures that each output is matched to at most one counterpart in the other list. When $m \neq n$, only $\min(m, n)$ outputs from each list are matched; the remaining $|m - n|$ outputs are left unmatched and do not contribute to the valid match count.

We then count *valid matches* that exceed a similarity threshold θ:

$$VM = \sum_{i=1}^{\min(m,n)} \mathbb{I}[\mathrm{sim}(s_{1,i}, s_{2,\pi(i)}) \geq \theta], \tag{5.10}$$

where $\mathbb{I}(\cdot)$ is the indicator function, returning 1 if the condition is true and 0 otherwise.

The corresponding precision, recall, and F1 score are:

$$P_{\mathrm{bip}} = \frac{VM}{m}, \quad R_{\mathrm{bip}} = \frac{VM}{n}, \quad F1_{\mathrm{bip}} = \begin{cases} \dfrac{2 \cdot P_{\mathrm{bip}} \cdot R_{\mathrm{bip}}}{P_{\mathrm{bip}} + R_{\mathrm{bip}}}, & \text{if } P_{\mathrm{bip}} + R_{\mathrm{bip}} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}$$

Here, $F1_{\text{bip}} \in [0, 1]$ with 1 indicating perfect one-to-one semantic alignment and 0 indicating no valid matches above θ. Compared to Soft-F1, this metric enforces a strict one-to-one correspondence, providing a more conservative evaluation of semantic agreement.

4. Average Cosine Similarity (Avg-Cosine). As a complementary measure, we compute the mean pairwise cosine similarity between all embeddings from the two output lists:

$$\text{AvgCos}(L_{\text{target}}, L_{\text{ref}}) = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \text{sim}(s_{1,i}, s_{2,j}), \tag{5.12}$$

where $\text{sim}(s_{1,i}, s_{2,j})$ denotes the cosine similarity between embeddings.

Cosine similarity lies in the range $[-1, 1]$. Although sentence encoders often yield values concentrated in a narrower region (often near $[0, 1]$), we preserve the theoretical bounds and apply the linear mapping:

$$\text{AvgCos}_{[0,1]} = \frac{\text{AvgCos} + 1}{2},$$

which transforms scores to $[0, 1]$ while maintaining relative distances. Higher $\text{AvgCos}_{[0,1]}$ values indicate stronger overall semantic similarity between the two output lists.

In contrast to Soft-F1 and Bipartite-F1, which rely on threshold-based alignment and impose explicit many-to-many or one-to-one matching structures, $\text{AvgCos}_{[0,1]}$ captures continuous semantic proximity across all pairwise combinations without enforcing any matching constraints. This provides a holistic similarity signal that is less sensitive to threshold selection and complements alignment-based metrics by reflecting global semantic relatedness between the two output sets. One limitation is that all pairs contribute equally, so redundant or loosely related outputs can bias the average similarity score by introducing noise or diluting the discriminative power of the metric.

5.2.3 Holistic Consensus-based Evaluation

To provide a more comprehensive evaluation, we establish a cross-model semantic consensus and assess the target system against this consensus baseline. This holistic two-stage framework comprises: (1) establishing cross-model semantic consensus through multi-threshold clustering, and (2) evaluating the target system against this consensus using semantic agreement metrics (addressing RQ 1 and RQ 2). Critically, the target system is excluded from consensus formation in Stage 1 to ensure an independent reference baseline.

The clustering threshold τ controls the semantic granularity of consensus formation, while the similarity threshold θ determines valid matches in the Soft-F1 and Bipartite-F1 evaluations. Integrating both stages enables a robust analysis that goes beyond direct pairwise comparisons by first constructing a reliable consensus baseline and then quantifying the target system's semantic alignment using complementary evaluation metrics. The complete procedure is summarized in Algorithm 4 and illustrated in Figure 5.1.

5.2.3.1 Stage 1: Consensus Establishment via Semantic Clustering

The first stage establishes a shared semantic baseline through clustering across multiple thresholds. For each document, outputs generated by all consensus models are aggregated (Figure 5.1 ①) and converted into dense vector embeddings (Figure 5.1 ②).

Agglomerative clustering (Figure 5.1 ③) is applied to group semantically coherent concepts (Figure 5.1 ④). Clustering is performed over a range of distance thresholds $\mathcal{T}$ to ensure robustness. Within each cluster C_k containing n_k embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_{n_k}\}$, the cluster centroid is computed as the mean embedding:

$$\mathbf{c}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{e}_i. \tag{5.13}$$

The canonical representative for the cluster is selected as the most frequent output. In the event of a tie, the candidate whose embedding $\mathbf{e}_i$ is closest to the centroid by cosine similarity is chosen:

$$s^* = \arg\max_{s_i \in C_k} \text{sim}(\mathbf{e}_i, \mathbf{c}_k), \tag{5.14}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity. This procedure ensures that each cluster is represented by a semantically central, frequently occurring output (`GetClusterRepresentatives` in Algorithm 4).

Consensus reliability is quantified using Krippendorff's Alpha, computed on nominal conceptual labels derived from clustering (Figure 5.1 ⑤). For each threshold τ and document d, each model m's output s_i^m is assigned a cluster label $\ell_i^m(\tau)$ corresponding to the cluster it belongs to after agglomerative clustering. Krippendorff's Alpha is then computed on the resulting label matrix, where rows correspond to output positions and columns correspond to models, measuring how consistently models assign outputs to the same conceptual clusters.

Importantly, Alpha is computed globally across all documents for each threshold τ, pooling cluster labels from all documents to produce a single α_τ value. This ensures that reliability reflects cross-document consistency rather than averaging potentially unstable per-document estimates. The final reliability score, α_{final}, is obtained by averaging α values across all thresholds (Figure 5.1 ⑥). We report per-threshold $\alpha(\tau)$ with 95% bootstrap confidence intervals, as well as α_{final} with a 95% bootstrap confidence interval computed by resampling documents and averaging $\alpha(\tau)$ within each draw.

The consensus output list at threshold τ is denoted as $L_{\text{consensus}}(\tau) = \{\hat{s}_k(\tau)\}_{k=1}^{K_\tau}$, where K_τ is the number of clusters and each $\hat{s}_k(\tau)$ is the canonical (most frequent) output within its cluster. This list serves as the reference baseline for evaluating the target system.

For each document d, outputs from all models are pooled as:

$$S_d = \bigcup_{m \in \{\text{LLMs}\}} S_d^m. \tag{5.15}$$

Agglomerative clustering uses cosine distance with average linkage:

$$d_{\text{cosine}}(E(s_i), E(s_j)) = 1 - \text{sim}(s_i, s_j), \tag{5.16}$$

$$d_{\text{avg}}(C_p, C_q) = \frac{1}{|C_p| \cdot |C_q|} \sum_{s_i \in C_p} \sum_{s_j \in C_q} d_{\text{cosine}}(E(s_i), E(s_j)). \tag{5.17}$$

Thresholds τ are selected to span a range from fine-grained to coarse-grained semantic groupings.

Consensus Reliability. Krippendorff's Alpha quantifies the reliability of the consensus clustering by measuring how consistently different models assign outputs to the same conceptual clusters. It evaluates the extent to which agreement among models exceeds what would be expected by chance.

Formally, for each threshold configuration, Krippendorff's Alpha is defined as:

$$\alpha = 1 - \frac{D_o}{D_e}, \tag{5.18}$$

where D_o denotes the *observed disagreement* among models, reflecting how often their output assignments differ, and D_e represents the *expected disagreement* under statistical independence (i.e., if models were labeling randomly).

An α value of 1 indicates perfect consensus, while $\alpha = 0$ implies agreement no better than chance. Negative values indicate systematic disagreement. To obtain a robust reliability estimate across clustering granularities, the final reliability score is averaged across all thresholds:

$$\alpha_{\text{final}} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \alpha_{\tau}. \tag{5.19}$$

This averaging accounts for variations in cluster granularity and ensures that the consensus reliability reflects stable semantic agreement across multiple levels of abstraction rather than depending on a single threshold configuration.

The resulting $L_{\text{consensus}}(\tau)$ and α_{final} define the consensus baseline for target system evaluation.

Once the consensus clusters and reliability are established, the second stage measures how closely the target system's output structures align with this consensus. This ensures evaluation is grounded in a cross-model, semantically validated reference rather than a single model's perspective.

5.2.3.2 Stage 2: Target System Evaluation Against Consensus

In the second stage, the target system's outputs are evaluated against the consensus baseline using semantic agreement metrics (addressing RQ 2). For each document, target system-generated outputs (Figure 5.1 ⑦) are embedded (Figure 5.1 ⑧) and compared with the consensus outputs from Stage 1 (Figure 5.1 ⑨).

Two complementary agreement scores are computed for each document d and threshold τ: Soft-F1 and Bipartite-F1 (Figure 5.1 ⑩, ⑪). Final scores are obtained by first averaging document-level results for each threshold, then averaging across thresholds (Figure 5.1 ⑫-⑬).

A. Soft-F1 Evaluation. Target system outputs are compared with the consensus outputs $L_{\text{consensus}}(\tau)$ using the Soft-F1 metric (Equations 5.5-5.7). For each threshold τ and document d, we compute $F1^{d}_{\text{soft}}(\tau)$. The final score averages across documents and thresholds:

$$\text{Soft-F1}_{\text{final}} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \left[\frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} F1^{d}_{\text{soft}}(\tau) \right]. \tag{5.20}$$

B. Bipartite Matching Evaluation. Target system outputs are optimally aligned with consensus outputs $L_{\text{consensus}}(\tau)$ using the Hungarian algorithm (Equations 5.8-5.11). Bipartite-F1 scores are computed for each document and threshold, then averaged:

$$\text{Bipartite-F1}_{\text{final}} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \left[\frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} F1^{d}_{\text{bip}}(\tau) \right]. \tag{5.21}$$

Both metrics are computed per document and averaged across thresholds to produce consensus-validated agreement scores. Soft-F1 captures many-to-many semantic overlap, while Bipartite-F1 enforces strict one-to-one correspondence, offering complementary perspectives on the target system's semantic alignment.

Algorithm 4 Cross-Model Semantic Evaluation Framework

Require:
Document set $\mathcal{D}$
Consensus models $\mathcal{M} = \{\text{Claude Sonnet 4, GPT-4o-mini, Google/Gemma-27B-IT}\}$
Target system
Clustering thresholds $\mathcal{T} = \{0.3, 0.35, 0.4, 0.45, 0.5\}$
Ensure: Cross-model reliability α_{final}, target system semantic agreement scores

Stage 1: Consensus Establishment
for $d \in \mathcal{D}$ **do**
  $S_d \leftarrow \bigcup_{m \in \mathcal{M}} S_d^m$ ▷ Pool outputs from all models (Eq. 5.15)
  $E_d \leftarrow \{E(s) : s \in S_d\}$ ▷ Compute embeddings (Eq. 5.1)
  for $\tau \in \mathcal{T}$ **do**
    $C_d(\tau) \leftarrow \text{AgglomerativeClustering}(E_d, \tau)$ ▷ Cosine distance + average linkage (Eqs. 5.16-5.17)
    Collect cluster labels $\ell_i^m(\tau)$ for all models $m \in \mathcal{M}$ ▷ Assign conceptual labels per model
  end for
end for
for $\tau \in \mathcal{T}$ **do**
  $\alpha_\tau \leftarrow \text{KrippendorffAlpha}(\{\ell_i^m(\tau) : \forall d, m\})$ ▷ Global reliability across all documents (Eq. 5.18)
end for
$\alpha_{\text{final}} \leftarrow \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \alpha_\tau$ ▷ Average across thresholds (Eq. 5.19)
Stage 2: Target System Evaluation
for $d \in \mathcal{D}$ **do**
  $S^d_{\text{target}} \leftarrow \text{TargetSystem}(d)$ ▷ Generate target system outputs for document d
  $E^d_{\text{target}} \leftarrow \{E(s) : s \in S^d_{\text{target}}\}$ ▷ Compute embeddings for target system outputs
  for $\tau \in \mathcal{T}$ **do**
    $S^d_{\text{consensus}}(\tau) \leftarrow \text{GetClusterRepresentatives}(C_d(\tau))$ ▷ Canonical outputs per cluster
    $F1^d_{\text{soft}}(\tau) \leftarrow \text{SoftF1}(E^d_{\text{target}}, S^d_{\text{consensus}}(\tau))$ ▷ Many-to-many matching (Eq. 5.7)
    $F1^d_{\text{bip}}(\tau) \leftarrow \text{BipartiteF1}(E^d_{\text{target}}, S^d_{\text{consensus}}(\tau))$ ▷ One-to-one matching (Eq. 5.11)
  end for
end for
$\text{Soft-F1}_{\text{final}} \leftarrow \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} F1^d_{\text{soft}}(\tau)$ ▷ Average Soft-F1 (Eq. 5.20)
$\text{Bipartite-F1}_{\text{final}} \leftarrow \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} F1^d_{\text{bip}}(\tau)$ ▷ Average Bipartite-F1 (Eq. 5.21)

In contrast to direct pairwise evaluations such as Soft-F1, Bipartite-F1, or Avg-Cosine, which measure similarity between individual systems, the proposed consensus-based framework introduces an intermediate semantic baseline that aggregates outputs across multiple models. This baseline acts as a shared, model-agnostic reference that mitigates bias arising from asymmetric comparisons and model-specific representations. By clustering semantically related outputs and validating their reliability through Krippendorff's Alpha, the framework captures a higher-order semantic agreement that direct pairwise methods cannot fully reflect. Evaluating the target system against this consensus therefore provides a more stable and interpretable assessment of cross-model semantic alignment.

5.3 Experiments

We evaluate the proposed framework using CIR3 as the target system through two complementary studies: **Study A (Cross-domain; Single Encoder)**, which examines robustness across dataset domains using a single embedding model, and **Study B (Cross-encoder; In-domain)**, which investigates sensitivity to embedding backbones within a single domain.

Both studies employ two evaluation approaches (Section 5.2): (1) pairwise semantic agreement evaluation comparing CIR3 against individual reference models using Jaccard, Soft-F1, Bipartite-F1, and Avg-Cosine metrics (Section 5.2.2); and (2) holistic consensus-based evaluation, which establishes a consensus baseline through multi-threshold clustering (Stage 1) and evaluates CIR3 against this consensus using Soft-F1 and Bipartite-F1 (Stage 2; Section 5.2.3).

5.3.1 Dataset and Target System

The CIR3 classifier agent serves as the target system, processing documents to identify relevant subtopics. These outputs are evaluated against individual reference models (pairwise evaluation; Section 5.2.2) and consensus outputs (holistic evaluation; Section 5.2.3). Consistent with Chapter 4 (Section 4.3), we adopt the same datasets and preprocessing procedures.

5.3.2 Experimental Design

We conduct two complementary studies to assess cross-domain robustness and embedding-model sensitivity of the proposed framework:

- **Study A (Cross-domain, 400 docs):** Assesses CIR3 on finance and medical corpora (200 documents each, Chapter 4) to evaluate cross-domain robustness and semantic agreement under both pairwise and holistic consensus approaches (RQ 1, RQ 2).

- **Study B (Finance-only, 1,000 docs):** Investigates embedding-model sensitivity, on finance corpora (Chapter 4), by comparing two embedding backbones, `BGE-Large` and `AMLM-Fusion`, for semantic alignment (RQ 3). `BGE-Large` is chosen for its strong performance and demonstrated cross-domain generalization.

All other experimental settings are held constant across both studies, including data preprocessing, consensus models, distance thresholds $\mathcal{T}$, and similarity threshold θ.

5.3.3 Consensus Models

Three diverse LLMs are employed as consensus models to capture varied semantic perspectives:

- **GPT-4o-mini** [40]: OpenAI's efficient variant providing rapid inference with strong reasoning capabilities.
- **Gemma-3-27B-it** [38]: Google's instruction-tuned model offering alternative architectural perspectives.
- **Claude Sonnet 4** [39]: Anthropic's advanced model with distinct training methodology and capabilities.

Cumulatively, this set spans distinct training paradigms and alignment practices (OpenAI, Google, Anthropic), which improves consensus robustness by aggregating perspectives that differ in data curation, safety alignment, and decoding behavior. CIR3 outputs are excluded from consensus establishment in Stage 1. Consensus generation uses fixed decoding settings: temperature $= 0.1$, top-$p = 0.5$.

5.3.4 Implementation Parameters

- **Clustering:** Agglomerative clustering with cosine distance and average linkage.
- **Thresholds:** Multi-threshold evaluation with $\tau \in \{0.30, 0.35, 0.40, 0.45, 0.50\}$[1] and semantic matching threshold $\theta = 0.7$.
- **Reliability:** Krippendorff's Alpha computed via `simpledorff` [229] library.

[1] Lower τ values tend to fragment semantically related concepts, whereas higher values risk conflating distinct ones. The range $\tau \in \{0.30, 0.35, 0.40, 0.45, 0.50\}$ provides an effective compromise in semantic granularity and is validated through preliminary experiments.

- **Evaluation Metrics:** Pairwise evaluation uses Jaccard, Soft-F1, Bipartite-F1, and Avg-Cosine. Holistic evaluation uses Soft-F1 and Bipartite-F1, both averaged across thresholds.
- **Reproducibility:** Random seeds are fixed for Python's `random`, NumPy, and PyTorch.
- **Sensitivity:** We probe $\theta \in [0.60, 0.80]$ and observe stable trends; main tables use $\theta = 0.7$.

TABLE 5.1: Experimental Setup for Studies A and B.

	Study A	**Study B**
Documents	400	1,000
Domain	Finance / Medical	Finance
Embedding	BGE (1024-d)	BGE-Large / AMLM-Fusion
	Shared Parameters	
Consensus Models	GPT-4o-mini, Gemma-3-27B-it, Claude Sonnet 4	
Clustering Thresholds (τ)	{0.30, 0.35, 0.40, 0.45, 0.50}	
Similarity Threshold (θ)	0.7	
Stage 2 Metrics	Soft-F1, Bipartite-F1	
Reliability (Stage 1): Krippendorff's Alpha	α averaged over $\mathcal{T}$	

Table 5.1 summarizes the experimental parameters for both studies.

TABLE 5.2: Hardware and Software Used for Experiments.

Component	**Specification**	**Component**	**Specification**
GPU	NVIDIA GeForce RTX 3080 CUDA Cores: 6144 Memory bandwidth: 448.06 GB/s VRAM: 16384 MB GDDR6 Shared system memory: 16265 MB CUDA 12.8 cuDNN 9.7.1	CPU	Intel Core i9-10980HK 8 cores / 16 threads 2.40 GHz, x86_64
		RAM	16.65 GB
		OS	Debian GNU/Linux 12 (bookworm)

Hardware and software environment parameters are listed in Table 5.2.

5.3.5 Statistical Analysis

Document-level agreement scores are analyzed using paired two-sided t-tests. To assess concordance between matching strategies, we compute Pearson correlations between Soft-F1 and Bipartite-F1 scores, with Spearman ρ computed as a rank-based robustness check. In Study B, we additionally compute document-level correlations between BGE-Large and AMLM-Fusion agreement scores to evaluate embedding-model sensitivity.

We report 95% confidence intervals alongside mean ± standard deviation where applicable. For agreement means (Soft-F1, Bipartite-F1), we use t-based confidence intervals with $\mathrm{df} = N - 1$:

$$\mathrm{CI}_{95} = \bar{x} \pm t_{0.975,\, N-1} \frac{s}{\sqrt{N}},$$

where $\bar{x}$ is the sample mean, s the sample standard deviation, and N the number of documents. Krippendorff's α is reported as a point estimate averaged across thresholds.

5.3.6 Interpretation Guidelines

- **Consensus Reliability:** We follow Krippendorff's guidelines [157, 230]: $\alpha \geq 0.800$ indicates high reliability; $0.667 \leq \alpha < 0.800$ permits tentative conclusions; $\alpha < 0.667$ indicates insufficient reliability.
- **CIR3 Performance:** High Soft-F1 and Bipartite-F1 scores indicate strong alignment with consensus.
- **Embedding Robustness:** High consistency between BGE-Large and AMLM-Fusion demonstrates that the framework remains stable across different encoder backbones, consistent with observations from Chapter 3 (Sections 3.3.3.1 and 3.4.2).
- **Threshold Sensitivity:** Multi-threshold evaluation provides insight into optimal clustering parameters.
- **Metric Concordance:** Correlations between pairwise and holistic metrics validate complementary perspectives.

5.4 Results

We report results from both studies. Study A examines consensus reliability across domains and compares pairwise versus holistic evaluation approaches, addressing RQ 1 and RQ 2. Study B evaluates embedding robustness within the finance domain, addressing RQ 3. Agreement computations use the same sentence encoder employed to derive $L_{\text{consensus}}(\tau)$.

5.4.1 Study A: Cross-domain Evaluation

TABLE 5.3: Cross-Domain Similarity Agreement Metrics for Subtopic Identification (N=200 per domain).

Metric	**FiQA/InsurQA**	**MedQA/MedMCQA**
Pairwise Semantic Agreement (Mean ± Std)		
Jaccard	0.5530 ± 0.2819	0.5403 ± 0.2938
Soft-F1	0.9161 ± 0.1060	0.9157 ± 0.0993
Bipartite	0.8388 ± 0.1596	0.8198 ± 0.1538
Avg-Cosine	0.7093 ± 0.0368	0.7074 ± 0.0332
Holistic Semantic Agreement		
Cross-Model Reliability (α)	0.9338	0.9207
CIR3 Semantic Agreement	0.9434 ± 0.0491 [95% CI: 0.9366–0.9502]	0.9316 ± 0.0473 [95% CI: 0.9250–0.9382]

95% CIs computed from sample mean and standard deviation using the t critical value with df = N−1 (N as specified for each study/domain).

Table 5.3 presents comprehensive evaluation results across both pairwise and holistic metrics.

5.4.1.1 Pairwise Semantic Agreement

Pairwise semantic agreement metrics demonstrate strong performance across both domains. Soft-F1 scores of 0.9161 (finance) and 0.9157 (medical) indicate strong many-to-many semantic alignment. Bipartite-F1 scores of 0.8388 (finance) and 0.8198 (medical) provide more conservative one-to-one matching evaluation. Jaccard scores are lower (0.5530 and 0.5403), reflecting the stricter nature of this metric in requiring exact token overlap.

5.4.1.2 Holistic Consensus-Based Evaluation

Holistic consensus-based evaluation (addressing RQ 1) demonstrates strong reliability, with Krippendorff's Alpha of 0.9338 (finance) and 0.9207 (medical) indicating robust inter-model agreement. Averaging α across thresholds reduces sensitivity to clustering granularity.

CIR3 achieves high semantic agreement with the consensus, with Soft-F1 scores of 0.9434 (finance) and 0.9316 (medical), demonstrating strong alignment with the consensus baseline (addressing RQ 2) and supporting the robustness of the holistic consensus-based evaluation framework. The slightly higher agreement in finance likely reflects its more controlled, domain-specific vocabulary. Overall, these results show that the framework produces stable, reliable semantic baselines across domains while CIR3 consistently aligns with the multi-model consensus.

5.4.1.3 Document-Level Performance Distribution

To assess consistency in subtopic prediction, we analyze the distribution of Bipartite-F1 scores across 400 documents. Bipartite-F1 was chosen for its higher standard deviation (Table 5.3), making it particularly sensitive to performance variation and a suitable measure for evaluating classifier agreement.

TABLE 5.4: Statistics of Classifier Bipartite F1 Scores across 400 Documents.

Score Interval	Doc Count (%)	Avg Score	Median	Min	Max	CIR3 Comp	CIR3 Faith
(0.50, 0.66]	4	0.600	0.600	0.600	0.600	0.880	0.923
(0.66, 0.75]	22	0.724	0.733	0.667	0.733	0.918	0.958
(0.75, 1.00]	74	0.875	0.867	0.800	1.000	0.949	0.994

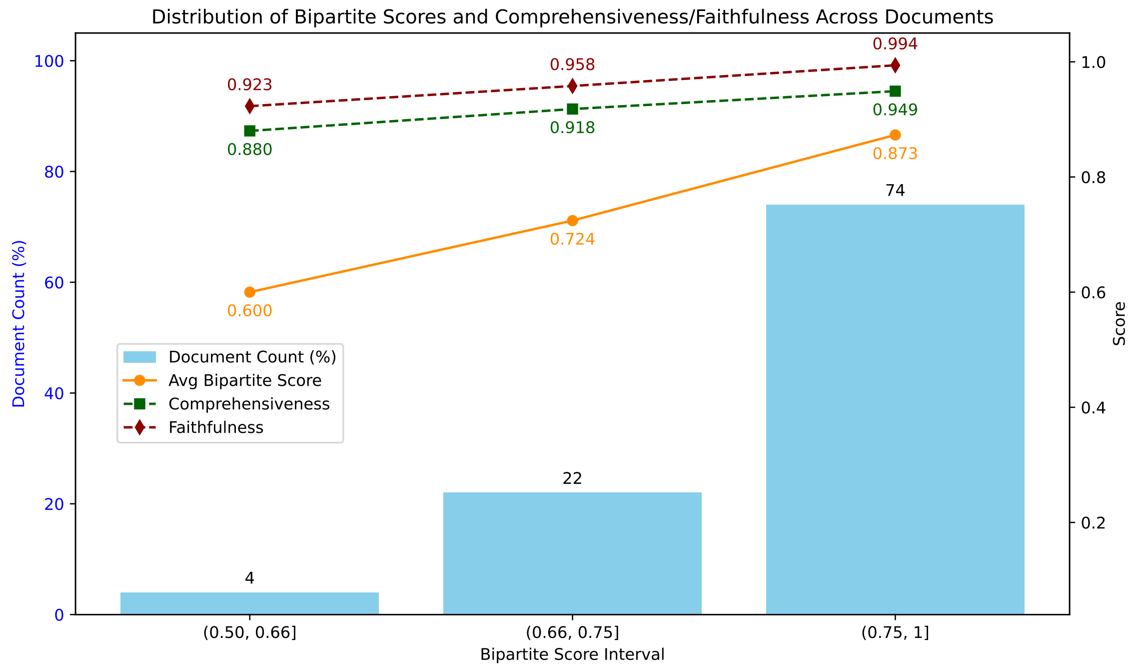

FIGURE 5.2: Study A (N=400): Distribution of 400 documents (200 finance, 200 medical) across Bipartite-F1 intervals for CIR3.

Analysis of Study A (Table 5.4, Figure 5.2) shows strong semantic alignment between CIR3 and the reference models: 74% of documents fall in the highest agreement interval $(0.75, 1]$, 22% in $(0.66, 0.75]$, and only 4% ≤ 0.66, with no document below 0.60. This indicates stable classifier behavior and a practical lower bound for one-to-one semantic agreement.

System-level Comprehensiveness and Faithfulness increase with Bipartite-F1, reaching 0.949 and 0.994, respectively, in the top interval. This correlation demonstrates that documents with higher classifier agreement corresponds to stronger end-task performance, providing indirect extrinsic validation of CIR3 and supporting the reliability of the holistic consensus-based evaluation framework. The framework thus produces meaningful, stable semantic baselines aligned with downstream system performance of CIR3.

5.4.2 Study B: Embedding Robustness Evaluation

Study B evaluates embedding robustness by comparing BGE-Large and AMLM-Fusion on 1,000 finance documents (addressing RQ 3), holding all other experimental settings constant (datasets, consensus models, distance thresholds $\mathcal{T}$, and similarity threshold θ).

TABLE 5.5: Finance (N=1,000): Similarity Agreement Metrics by Embedding Backbone.

Metric	BGE-Large	AMLM-Fusion
Pairwise Semantic Agreement (Mean ± Std)		
Jaccard	0.5416 ± 0.2829	
Soft-F1	0.9119 ± 0.1028	0.9286 ± 0.1160
Bipartite	0.8293 ± 0.1547	0.8573 ± 0.1606
Avg-Cosine	0.6997 ± 0.0353	0.7080 ± 0.0603
Holistic Semantic Agreement		
Krippendorff's Alpha Reliability (α)	0.9413	0.9365
CIR3 Semantic Agreement (Soft-F1)	0.9434 ± 0.0491 [95% CI: 0.9404–0.9464]	0.9541 ± 0.0564 [95% CI: 0.9506–0.9576]
CIR3 Semantic Agreement (Bipartite F1)	0.8213 ± 0.1541 [95% CI: 0.8117–0.8309]	0.8449 ± 0.1595 [95% CI: 0.8350–0.8548]

95% CIs computed from sample mean and standard deviation using the t critical value with df = N−1 (N=1,000 documents per model).

Table 5.5 reports evaluation results across pairwise and holistic metrics, demonstrating consistent performance across embedding backbones.

5.4.2.1 Pairwise Semantic Agreement

Pairwise metrics indicate strong performance, with Soft-F1 scores of 0.9119 (BGE) and 0.9286 (AMLM-Fusion), and Bipartite-F1 scores of 0.8293 (BGE) and 0.8573 (AMLM-Fusion). Soft-F1 reflects the flexibility of many-to-many matching, while Bipartite-F1 provides a more conservative one-to-one evaluation, capturing strict semantic alignment.

5.4.2.2 Holistic Consensus-Based Evaluation

Holistic consensus-based evaluation shows similarly strong agreement: Soft-F1 of 0.9434 (BGE) and 0.9541 (AMLM-Fusion), and Bipartite-F1 of 0.8213 (BGE) and 0.8449 (AMLM-Fusion). AMLM-Fusion demonstrates marginally higher performance in both pairwise and holistic metrics, suggesting slight benefits from incorporating domain-specific knowledge for the in-domain classification task.

Krippendorff's Alpha values of 0.9413 (BGE) and 0.9365 (AMLM-Fusion) confirm robust consensus reliability across embeddings, demonstrating that the holistic evaluation framework is resilient to different embedding strategies (addressing RQ 3). The consistent results highlight the framework's embedding robustness and reinforce its generalizability for in-domain evaluation.

5.4.2.3 Document-Level Performance Distribution (Finance)

Document-level analysis mirrors Study A. Figure 5.3 and Table 5.6 summarize the distribution of Bipartite-F1 scores for BGE-Large and AMLM-Fusion, along with corresponding CIR3 Comprehensiveness and Faithfulness scores. Most documents fall in the highest agreement interval $(0.75, 1]$, with Comprehensiveness and Faithfulness scores increasing alongside Bipartite-F1.

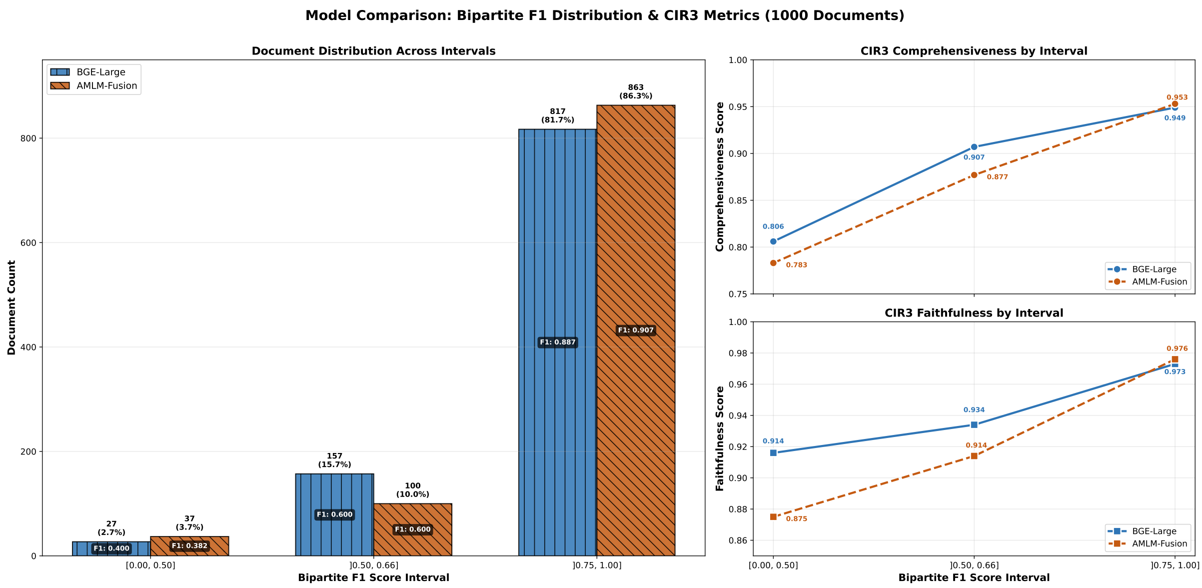

FIGURE 5.3: Study B: Distribution of 1,000 finance documents across Bipartite-F1 intervals under alternative embedding backbones (`BGE` vs. `AMLM-Fusion`).

TABLE 5.6: Statistics of Classifier Bipartite F1 Scores across 1000 Documents.

Model	**Score Interval**	**Doc Count (%)**	**Avg Bipartite F1**	**Median**	**Min**	**Max**	**CIR3 Comp**	**CIR3 Faith**
BGE-Large	[0, 0.50]	27 (2.7%)	0.400	0.400	0.400	0.400	0.806	0.916
	(0.50, 0.66]	157 (15.7%)	0.600	0.600	0.600	0.600	0.907	0.934
	(0.66, 0.75]	0 (0.0%)	0.000	0.000	0.000	0.000	-	-
	(0.75, 1.00]	817 (81.7%)	0.887	0.800	0.800	1.000	0.949	0.973
AMLM-Fusion	[0, 0.50]	37 (3.7%)	0.382	0.400	0.200	0.400	0.783	0.875
	(0.50, 0.66]	100 (10.0%)	0.600	0.600	0.600	0.600	0.877	0.914
	(0.66, 0.75]	0 (0.0%)	0.000	0.000	0.000	0.000	-	-
	(0.75, 1.00]	863 (86.3%)	0.907	1.000	0.800	1.000	0.953	0.976
Final Average							**0.939**	**0.966**

This correlation demonstrates that higher one-to-one agreement aligns with stronger end-task performance of CIR3, providing indirect, extrinsic validation of the holistic consensus-based framework. Overall, these results confirm that the framework produces stable, reliable semantic baselines that are robust to different embedding models and meaningfully reflect downstream system performance.

5.4.2.4 Metric Concordance and Embedding Robustness

Table 5.7 shows strong concordance between metrics across both encoders, indicating that Soft-F1 and Bipartite-F1 rank documents consistently while providing complementary perspectives.

TABLE 5.7: Metric concordance and embedding robustness in finance (N=1,000).

Comparison	**Pearson r**	**Spearman ρ**
Within model (BGE): Soft-F1 vs Bipartite F1	0.7064 ($p < 0.0001$)	0.6595 ($p < 0.0001$)
Within model (AMLM-Fusion): Soft-F1 vs Bipartite F1	0.7903 ($p < 0.0001$)	0.7277 ($p < 0.0001$)
Cross model: Soft-F1 (BGE vs AMLM-Fusion)	0.6634 ($p < 0.0001$)	0.6353 ($p < 0.0001$)
Cross model: Bipartite F1 (BGE vs AMLM-Fusion)	0.8117 ($p < 0.0001$)	0.7878 ($p < 0.0001$)

Cross-encoder correlations are similarly high, indicating that document-level conclusions remain stable regardless of embedding choice. The results in Table 5.7 show strong cross-model agreement for both metrics, with Pearson correlations of 0.6634 for Soft-F1 and 0.8117 for Bipartite-F1, and corresponding Spearman values of 0.6353 and 0.7878. This pattern is unlikely to be incidental. In Chapter 3, BGE-Large and AMLM-Fusion also showed strong correlations on an independent semantic similarity task (Sections 3.3.3.1 and 3.4.2), and the same encoders correlate strongly again within the consensus evaluation in this work. The fact that both chapters independently yield consistent cross-encoder agreement strengthens the validity of AMLM-Fusion itself while also providing additional evidence that the consensus-based evaluation framework is not sensitive to the underlying embedding backbone. Combined with the CIR3 findings in Chapter 4, these results offer broader extrinsic support for the stability and reliability of the holistic consensus evaluation method.

5.5 Conclusion and Future Work

This chapter introduced a consensus-based evaluation framework for generative semantic tasks that operates without human-annotated gold standards. The framework constructs a cross-model semantic baseline through multi-threshold clustering and evaluates system alignment using complementary agreement metrics. Together, these components provide a reliable alternative for settings where labeled data are unavailable.

Empirically, we addressed our research questions. First, we observed strong consensus reliability across thresholds (finance/medical alpha near 0.93/0.92), indicating stable inter-model agreement (RQ 1). Second, evaluating the target system against the consensus produced high alignment (for instance, CIR3 Soft-F1 near 0.94 across domains), showing that the holistic approach offers interpretable evidence beyond pairwise matching (RQ 2). Third, robustness analyses demonstrated consistent behavior across embedding backbones: cross-model correlations between BGE-Large and AMLM-Fusion were high for both Soft-F1 and Bipartite-F1, and document-level performance was stable (no document below 0.60 in Study A), supporting encoder invariance and generalizability (RQ 3).

These findings also connect this chapter to prior contributions. CIR3 (Chapter 4) aligns strongly with the consensus baseline, providing extrinsic support for the quality of the consensus metrics themselves in low-resource settings. AMLM-Fusion (Chapter 3) performs on par with, and at times slightly above, BGE-Large while maintaining cross-model stability, indicating that domain-informed pretraining can be integrated without compromising evaluation reliability. The alignment between consensus metrics, downstream system performance, and cross-encoder consistency further reinforces the framework as a dependable method for evaluating semantic fidelity without gold labels.

Future work includes applying consensus-based evaluation to additional tasks such as summarization or data-to-text generation, studying active selection of consensus models, and exploring principled ways to weight model contributions within the consensus.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This dissertation set out to advance the semantic fidelity of language technologies in specialized domains such as finance and medicine. We define semantic fidelity (Chapter 1) as the degree to which a system preserves intended meaning and relations among domain concepts across three stages: (i) representation (capturing domain semantics), (ii) generation (producing outputs that are comprehensive and faithful to context), and (iii) evaluation (assessing meaning-preserving equivalence under lexical variation). The central challenge is that standard training and evaluation treat tokens and judgments uniformly, thereby reducing sensitivity to domain-critical semantics and making reliable assessment difficult when gold standards are scarce.

We introduced Adaptive Masked Language Modeling (AMLM; Chapter 3) as an approach to specialize models without architectural changes. AMLM shifts adaptation from input-side masking to output-side loss modulation, dynamically prioritizing domain terminology through weighted gradients Evaluated on financial domain tasks, AMLM achieves notable improvements: intrinsic dimensionality (kNN-MLE) reduced from 23.762 to 9.847, Pearson correlation improved by 0.188, and QA matching Recall@1 improved by 0.231. These results demonstrate that loss-side modulation effectively guides models toward compact, semantically coherent representations while maintaining training stability.

We then presented Collective Intentional Reading through Reflection and Refinement (CIR3; Chapter 4), a multi-agent framework for generating comprehensive and faithful QA pairs from information-dense documents. CIR3 operationalizes collective intelligence through three mechanisms: transactive reasoning (agents iteratively build on each other's insights), multi-perspective analysis (dynamic specialization by subtopic), and variation-guided convergence (maintaining diversity while reaching agreement). Experiments across finance and medical domains demonstrate substantial improvements: CIR3 achieves relative gains of 23% in comprehensiveness and 17% in faithfulness over strong LLM baselines according to automatic evaluation, with human evaluation confirming improvements of 15% and 11.66% respectively. Formalizing generation as a diversity-alignment objective provides a practical basis to balance coverage and faithfulness, reducing common errors such as under- / over-specifity, hallucination, irrelevance, and duplication.

Finally, we proposed a consensus-based evaluation framework (Chapter 5) that enables rigorous assessment in the absence of human-annotated gold standards. The framework establishes semantic consensus through hierarchical clustering across multiple models and granularities, quantifies inter-model reliability using Krippendorff's Alpha on induced concept labels, and evaluates target systems using complementary agreement metrics (Soft-F1 and Bipartite-F1). Validation across finance and medical domains demonstrates strong consensus reliability ($\alpha \approx 0.93$ for finance, $\alpha \approx 0.92$ for medical), with CIR3 achieving high agreement with the consensus baseline (Soft-F1 $\approx 0.94/0.93$). Embedding robustness studies confirm consistent performance across both general-purpose (BGE-Large) and domain-informed (AMLM-Fusion) encoders, providing a practical and reliable alternative when expert annotation is limited.

Taken together, these contributions offer an end-to-end perspective: AMLM supplies domain-aware representations; CIR3 orchestrates a principled, model-agnostic multi-agent workflow to generate comprehensive and faithful outputs (compatible with, but not dependent on, domain-aware encoders); and the consensus framework evaluates both representation quality and generation fidelity without relying on gold labels. The components are modular: they can be adopted independently or combined as a pipeline, depending on data, resource, and deployment constraints.

These contributions address the three research questions posed in Chapter 1. Regarding RQ1 (adaptive loss-side weighting for domain-specific semantics), AMLM demonstrates that dynamically weighting token-level losses yields measurable improvements in semantic tasks while maintaining training stability. For RQ2 (principles enabling comprehensive and faithful QA generation), CIR3 shows that structured multi-agent coordination balancing perspectival diversity with semantic alignment operationalizes collective intelligence for balanced generation. Addressing RQ3 (rigorous evaluation without gold standards), the consensus framework establishes that multi-model semantic agreement with explicit reliability quantification provides a practical alternative to expert annotation.

Two broader implications emerge. First, progress in specialized domains does not require primary reliance on architectural changes: reweighting learning signals, orchestrating agent interaction, and measuring semantic agreement can yield measurable improvements with practical computational costs. Second, collective intelligence principles adapted from the social sciences can productively structure language model collaboration, suggesting that systems design may complement model scaling as a source of future improvements.

6.2 Scope and Limitations

This dissertation focuses on textual modalities in two specialized domains: finance and medicine. While the methods are designed to be domain-agnostic in principle, empirical validation is limited to these domains. The choice of finance and medicine domains reflects their high semantic density, economic and social importance, and the availability of suitable corpora. Generalization to other specialized domains (e.g., law, scientific research) remains an important direction for future work.

6.2.1 Methodological Limitations

Several choices warrant explicit acknowledgment:

- *AMLM term identification.* The approach relies on curated domain glossaries to identify specialized terminology; such lexicons may lag behind emerging terminology and may not capture context-dependent meanings.
- *CIR3 dependency on base models.* The framework's performance ultimately depends on the capabilities of underlying language models; the optimal number, roles, and interaction topology of agents may vary by task and domain.

- *Consensus costs and assumptions.* Establishing consensus requires multiple diverse models and clustering across thresholds, which introduces computational costs. The framework assumes that model diversity leads to robust consensus, an assumption validated in this work. Further study across broader model families and tasks remains warranted.

- *Reproducibility and hyperparameters.* Training was conducted with limited random seeds due to computational constraints; some hyperparameters (e.g., AMLM weighting constants) were selected based on preliminary experiments rather than exhaustive search.

- *Evaluation breadth.* Evaluation primarily employs automatic metrics, with limited human assessment. Larger-scale human studies would further strengthen conclusions about faithfulness and comprehensiveness.

- *External validity.* Generalization beyond English and to other specialized domains (e.g., law, scientific research) requires further validation.

6.3 Future Work

Building on these contributions, several directions appear promising:

AMLM extensions. The loss-side weighting paradigm opens several research directions: (i) dynamic lexicon construction and context-aware term detection to address evolving terminology; (ii) principled hyperparameter selection (e.g., learning α, β, τ, κ) through meta-learning or adaptive methods; and (iii) combining AMLM with parameter-efficient fine-tuning (LoRA, Adapters) and adaptive masking policies to explore synergies between loss-side and input-side adaptation.

CIR3 enhancements. Multi-agent orchestration can be refined through: (i) automated topology and role selection for agents based on document characteristics and task requirements; (ii) task- and domain-adaptive convergence criteria that balance computational cost with generation quality; and (iii) integration with dynamic tool-use and retrieval-augmented workflows to ground generation in external evidence.

Evaluation advances. The consensus framework can be extended through: (i) active selection and weighting of consensus models based on domain expertise or past performance; (ii) application to other generative tasks (summarization, data-to-text generation, dialogue) where gold standards are scarce; and (iii) larger-scale human evaluation studies to calibrate automatic metrics and establish correlation with domain expert judgments.

Cross-chapter configurations. While not required for this dissertation's results, AMLM-based encoders can be explored as alternative embedding backbones for the Vendi diversity tool in CIR3 or as auxiliary evaluation encoders alongside BERTScore and BGE. Such experiments would further probe interoperability while preserving the modularity of the contributions.

6.4 Closing Remarks

Advancing semantic fidelity in specialized domains remains central to deploying language models where precision matters. This dissertation contributes a pathway for achieving this goal: reweight learning signals to prioritize domain semantics, organize agent interaction to balance diversity and convergence, and evaluate through semantic consensus with explicit reliability. These ideas may be useful not only for the specific methods introduced in this work, but also as general design principles for building reliable NLP systems when data are scarce, stakes are high, and meaning is central.

Appendix A

AMLM

A.1 Naive Domain Token Identification Algorithm

For comparison, we include the naive nested-scan approach that illustrates the conceptual logic but has higher computational complexity:

Algorithm 5 Naive Domain Token Identification

Require:

Tokenized sequence $\mathbf{z} = (z_1, z_2, \dots, z_{L'})$
Glossary $\mathcal{G}$
Encoding function τ
Special tokens $\mathcal{S}$

Ensure: Domain token indicator $I_{\mathcal{G}}$, Match set $\mathcal{I}(\mathbf{z}, \mathcal{G})$

Initialize $\mathcal{I}(\mathbf{z}, \mathcal{G}) \leftarrow \emptyset$
Initialize $I_{\mathcal{G}}(i) \leftarrow 0$ for all $i \in [1, L']$
for each glossary term $g_j \in \mathcal{G}$ **do**
    Get encoded term $\tau(g_j) = (v_1^{(j)}, \dots, v_{k_j}^{(j)})$
    for each position $i \in [1, L' - k_j + 1]$ **do**
        if $\mathbf{z}_{[i, i+k_j)} = \tau(g_j)$ **and** $z_r \notin \mathcal{S}$ for $r \in [i, i + k_j)$ **then**
            Add (i, g_j) to $\mathcal{I}(\mathbf{z}, \mathcal{G})$
            for $r = i$ to $i + k_j - 1$ **do**
                Set $I_{\mathcal{G}}(r) \leftarrow 1$
            end for
        end if
    end for
end for
return $I_{\mathcal{G}}$, $\mathcal{I}(\mathbf{z}, \mathcal{G})$

This naive approach has complexity $\mathcal{O}(|\mathcal{G}| \times L' \times k)$, where $|\mathcal{G}|$ is the glossary size, L' the tokenized sequence length, and k the average term length. While conceptually clear, it becomes inefficient for large glossaries, motivating the Aho-Corasick implementation [173] in Algorithm 1.

A.2 Methodology: AMLM Conceptual Framework

Preprocessing (one-time)

Encode glossary terms: $\tau(g_j)$ for all $g_j \in \mathcal{G}$. Compute $\mathrm{IDF}(t)$ across the corpus if using TF-IDF.

For each sequence $X \in \mathcal{D}$: tokenize $\mathbf{z}$; identify domain tokens via Algorithm 1 with special-token filtering (e.g., [CLS], [SEP], [PAD]) and longest-match resolution; compute per-token weights $\mathbf{w}$ per chosen strategy; cache $\mathbf{w}$ in the dataset.

Training loop (per batch)

Apply MLM masking: $(\mathbf{X}_{\text{masked}}, \mathbf{Y}, \mathbf{M}) \leftarrow \text{mlm\_mask}(\mathcal{B})$.

Retrieve cached weights: $\mathbf{W} \leftarrow \text{get\_weights}(\mathcal{B})$. If using AMLM-Batch, compute batch stats (e.g., σ_B) over jargon counts.

Forward pass: logits $\mathbf{P} \leftarrow f_\theta(\mathbf{X}_{\text{masked}})$.

Per-token loss (no reduction): $\boldsymbol{\ell} \leftarrow \text{CrossEntropy}(\mathbf{P}, \mathbf{Y}, \texttt{reduction='none'})$. Unmasked positions (label -100) contribute zero.

Stabilize weights: clip to $[w_{\min}, w_{\max}]$; apply temperature smoothing (Equation 3.24); mix to meet ESS target (Equations 3.26–3.28) to obtain normalized $\tilde{\mathbf{W}}$.

Final weighted loss and update: $\mathcal{L} = \sum_{i \in M} \tilde{w}_i \, \ell_i$; update θ with AdamW and scheduling.

FIGURE A.1: AMLM end-to-end procedure aligned with Algorithm 2. The process consists of two phases. **Preprocessing (one-time)**: encodes glossary terms, computes corpus statistics (IDF), identifies domain tokens, and caches per-token weights. **Training loop (per batch)**: applies MLM masking, retrieves cached weights (and batch stats for AMLM-Batch), computes per-token losses with `reduction='none'`, stabilizes the weights, aggregates to a weighted loss, and updates parameters.

A.3 Integrating Focal Modulation into AMLM Strategies

The AMLM framework is modular, allowing focal-style modulation [68] to be integrated with the contextual weighting strategies described in Section 3.2.5. This is achieved by multiplicatively combining the contextual weight with the focal term.

Let w_i^{context} be the weight calculated by any of the contextual strategies (e.g., AMLM-Seq, AMLM-Batch, AMLM-TFIDF, AMLM-Fusion). The combined weight, w_i^{focal}, would then be:

$$w_i^{\text{focal}} = w_i^{\text{context}} \cdot \left(1 - P(x_i \mid X_{\setminus M}; \theta)\right)^{\gamma} \tag{A.1}$$

This only applies to the boosted (jargon) tokens where $s_i = 1$. The background tokens would retain their original contextual weight.

For our most comprehensive strategy, AMLM-Fusion, the weight for a jargon token ($I_{\mathcal{G}}(i) = 1$) would be calculated as:

$$w_i = \min\left(\alpha \cdot \text{boost}_X \cdot (1 + \text{TF-IDF}) \cdot (1 - P)^{\gamma}, w_{\max}\right) \tag{A.2}$$

The background token weight would remain $w_{\text{background}} = \max(\beta/\text{boost}_X, w_{\min})$.

This combined raw weight would then be passed to the stabilization layer (Section 3.2.8), which would apply clipping, temperature smoothing, and ESS targeting as usual. This demonstrates how the framework can synergistically combine linguistic importance (from the contextual strategies) with performance-based importance (from the focal term) in a robust and modular fashion. A systematic ablation of the focal-style option is left for future work.

A.4 Semantic Similarity Evaluation: Mathematical Details

To quantitatively assess how well each model captures fine-grained semantic relationships, we evaluate it on the **FinLang/investopedia-embedding-dataset**. Scores are generated for 22,940 QA pairs and evaluated against a set of high-quality reference scores from bge-large-en-v1.5. The alignment between model scores and the reference scores is measured using multiple metrics to provide a comprehensive view of performance. We report Pearson correlation (r) for linear relationships, alongside two rank-based metrics, Spearman's (ρ) and Kendall's (τ), to evaluate monotonic agreement. To measure absolute score calibration, we also include Mean

Squared Error (MSE), and Mean Absolute Error (MAE). We report 95% confidence intervals (CIs) for all correlation metrics to ensure comparisons are statistically meaningful.

A.4.1 Data and Notation

Let $\mathcal{D} = \{(q_i, a_i, y_i)\}_{i=1}^{N}$ denote the evaluation set of N paired observations (here, $N = 22{,}940$), where q_i is a query sentence, a_i its paired answer sentence, and $y_i \in \mathbb{R}$ is a provided reference similarity score (from `bge-large-en-v1.5`). For a model M, let $E_M(\cdot) \in \mathbb{R}^d$ denote the sentence embedding obtained by mean pooling the last hidden states.

A.4.2 Embedding and Cosine Similarity

For each pair (q_i, a_i) we compute embeddings

$$\mathbf{u}_i = E_M(q_i), \qquad \mathbf{v}_i = E_M(a_i).$$

We then apply L2 normalization

$$\hat{\mathbf{u}}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2}, \qquad \hat{\mathbf{v}}_i = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_2},$$

and define the model's cosine similarity score

$$s_i = \cos(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i) = \hat{\mathbf{u}}_i^\top \hat{\mathbf{v}}_i \in [-1, 1].$$

Collect the vectors $\mathbf{s} = (s_1, \ldots, s_N)$ and $\mathbf{y} = (y_1, \ldots, y_N)$.

A.4.3 Evaluation Metrics

We report correlation and error metrics between $\mathbf{s}$ and $\mathbf{y}$:

- **Pearson correlation** r: linear association

$$r = \frac{\sum_{i=1}^{N} (s_i - \bar{s})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N} (s_i - \bar{s})^2} \sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2}},$$

  where $\bar{s}$ and $\bar{y}$ are sample means.

- **Spearman's** ρ: Pearson correlation of the rank-transformed scores (ties handled by average ranks).
- **Kendall's** τ: based on the difference between the number of concordant and discordant pairs among all $\binom{N}{2}$ pairwise comparisons.
- **Mean Squared Error (MSE)**: $\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(s_i - y_i)^2$.
- **Mean Absolute Error (MAE)**: $\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|s_i - y_i|$.

A.4.4 Confidence Intervals

We quantify uncertainty for the correlation metrics with 95% confidence intervals (CIs) at level $\alpha = 0.05$.

- **Pearson** r **(Fisher transformation).** Let $z = \operatorname{atanh}(r)$ and $\mathrm{SE}(z) = 1/\sqrt{N-3}$. An approximate 95% CI on z is $[z - 1.96\,\mathrm{SE},\ z + 1.96\,\mathrm{SE}]$, which we map back to r via $\tanh(\cdot)$.
- **Spearman** ρ **and Kendall** τ **(bootstrap).** We draw $B = 2000$ bootstrap resamples by sampling indices $I^{(b)} \in \{1, \ldots, N\}^N$ with replacement, compute the statistic $T(\mathbf{s}_{I^{(b)}}, \mathbf{y}_{I^{(b)}})$ on each resample, and report the percentile CI $[\hat{T}_{\alpha/2}, \hat{T}_{1-\alpha/2}]$.

Determinism. Embedding generation is deterministic for a fixed model and inputs. Consequently, CIs reflect sampling variability over the N paired observations (via bootstrap), not variance across different training runs.

A.4.5 Procedure Summary

1. Compute mean-pooled embeddings for all q_i and a_i; apply L2 normalization.
2. Form cosine similarities $s_i = \hat{\mathbf{u}}_i^\top \hat{\mathbf{v}}_i$.
3. Compute metrics: r, ρ, τ, MSE, MAE, R^2.
4. Construct 95% CIs: Fisher for r; bootstrap (B=2000) for ρ and τ.

A.5 Embedding Space Analysis: Mathematical Details

To assess the intrinsic geometric properties of the learned representations, we conduct an analysis of the embedding space focusing on both representational efficiency and structural characteristics. This analysis is performed on the full set of 45,880 embeddings from the Investopedia dataset.

This section provides the formal definitions for the metrics used to analyze the intrinsic properties of the embedding space, as discussed in the main text.

A.5.1 Data and Notation

Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ be the set of N embeddings (here, $N = 45{,}880$), where each $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional vector obtained via mean pooling. Let $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$ be the global mean of the embeddings.

A.5.2 Intrinsic Dimensionality

We estimate the intrinsic dimensionality of the embedding manifold using the k-Nearest Neighbors Maximum Likelihood Estimator (kNN-MLE) [187]. For each point $\mathbf{x}_i$, let $d_k(\mathbf{x}_i)$ be the Euclidean distance to its k-th nearest neighbor. The estimator is given by:

$$\hat{d}_{\text{int}} = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{d_k(\mathbf{x}_i)}{d_j(\mathbf{x}_i)} \right)^{-1}$$

A lower estimated dimension $\hat{d}_{\text{int}}$ suggests a more compressed and specialized representation, indicating that the model has learned to represent the data in a lower-dimensional subspace that captures the essential structure of the financial domain.

A.5.3 Cluster Validity Indices

Both the Davies-Bouldin and Calinski-Harabasz scores rely on an initial k-means clustering of the embeddings $\mathcal{X}$ into a set of K clusters, $C = \{c_1, c_2, \ldots, c_K\}$. Let $|c_k|$ be the number of points in cluster k, and $\boldsymbol{\mu}_k$ be the centroid of cluster k. We use k-means++ initialization, the

Euclidean (L2) distance metric, and $n_{\text{init}} = 10$ restarts, selecting the solution with the best objective value (consistent with `scikit-learn` defaults).

A.5.3.1 Davies-Bouldin Index

The Davies-Bouldin index [188] measures the average "similarity" between each cluster and its most similar one, where similarity is a function of cluster size (scatter) and separation. For each cluster c_k, the average intra-cluster distance is $S_k = \frac{1}{|c_k|}\sum_{\mathbf{x}\in c_k}||\mathbf{x} - \boldsymbol{\mu}_k||_2$. The distance between two clusters is $d_{kj} = ||\boldsymbol{\mu}_k - \boldsymbol{\mu}_j||_2$. The Davies-Bouldin index is then:

$$DB = \frac{1}{K}\sum_{k=1}^{K}\max_{j\neq k}\left(\frac{S_k + S_j}{d_{kj}}\right)$$

A lower DB score indicates better clustering, with clusters being more compact and well-separated.

A.5.3.2 Calinski-Harabasz Score

Also known as the Variance Ratio Criterion [189], this score is the ratio of the between-cluster dispersion to the within-cluster dispersion. Let SS_W be the sum of squared distances within clusters and SS_B be the sum of squared distances between clusters:

$$SS_W = \sum_{k=1}^{K}\sum_{\mathbf{x}\in c_k}||\mathbf{x} - \boldsymbol{\mu}_k||_2^2, \qquad SS_B = \sum_{k=1}^{K}|c_k|||\boldsymbol{\mu}_k - \bar{\mathbf{x}}||_2^2$$

The Calinski-Harabasz score is defined as:

$$CH = \frac{SS_B/(K-1)}{SS_W/(N-K)}$$

A higher CH score indicates denser, better-separated clusters.

A.6 Financial QA Matching Evaluation: Mathematical Details

This evaluation tests the ability to match financial questions to their corresponding answers, assessing the semantic understanding of financial concepts rather than general information retrieval. For each benchmark, candidate passages comprise the full corpus specific to that dataset, and indexing uses brute-force similarity for exact evaluation.

This section provides the formal definitions for the Financial QA Matching evaluation methodology and the metrics used to assess performance, as discussed in the main text.

A.6.1 Setup and Notation

Let $\mathcal{D} = \{(q_i, p_i^*)\}_{i=1}^{N}$ denote the evaluation set of N query-passage pairs, where q_i is a query and p_i^* is its single ground-truth relevant passage. Let $\mathcal{P} = \{p_j\}_{j=1}^{M}$ be the full corpus of M candidate passages from which the relevant passage must be retrieved. For each query q_i, the candidate set is the entire corpus $\mathcal{P}$.

Let $E(\cdot) \in \mathbb{R}^d$ be the embedding function of a given model, which maps a text sequence to a d-dimensional vector. In our experiments, this is implemented by mean-pooling the last hidden-layer states of the transformer.

A.6.2 Embedding, Normalization, and Scoring

For each query q_i and each passage $p_j \in \mathcal{P}$, we first compute their respective embeddings:

$$\mathbf{u}_i = E(q_i), \qquad \mathbf{v}_j = E(p_j).$$

These embeddings are then L2-normalized to project them onto the unit hypersphere, ensuring that the similarity score is independent of vector magnitude:

$$\hat{\mathbf{u}}_i = \frac{\mathbf{u}_i}{\|\mathbf{u}_i\|_2}, \qquad \hat{\mathbf{v}}_j = \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|_2}.$$

The relevance score between query q_i and passage p_j is computed using cosine similarity:

$$s(q_i, p_j) = \cos(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_j) = \hat{\mathbf{u}}_i^\top \hat{\mathbf{v}}_j \in [-1, 1].$$

A.6.3 Ranking and Evaluation Metrics

For each query q_i, we generate a ranked list of all passages in $\mathcal{P}$ by sorting them in descending order based on their similarity score $s(q_i, p_j)$. Let rank_i be the position (rank) of the ground-truth passage p_i^* in this list. The following metrics are used to evaluate the quality of this ranking over the entire dataset of N queries.

A.6.3.1 Recall@K

Recall@K measures the fraction of queries for which the correct passage is ranked within the top K positions.

$$\text{Recall@K} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(\text{rank}_i \leq K),$$

where $\mathbf{1}(\cdot)$ is the indicator function.

A.6.3.2 Mean Reciprocal Rank (MRR)

MRR is the average of the reciprocal of the rank of the correct passage. It heavily penalizes rankings where the correct item appears late in the list.

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}.$$

A.6.3.3 Mean Average Precision (MAP)

MAP provides a single-figure measure of quality across all recall levels. For a single query, Average Precision (AP) is defined as:

$$\text{AP}_i = \frac{1}{G_i} \sum_{k=1}^{M} P_i(k) \times \text{rel}_i(k),$$

where G_i is the total number of relevant documents (in our case, $G_i = 1$), $P_i(k)$ is the precision at rank k, and $\text{rel}_i(k)$ is an indicator function that is 1 if the item at rank k is relevant and 0 otherwise. Since we have only one relevant document, this simplifies to $\text{AP}_i = 1/\text{rank}_i$. MAP is the mean of the AP scores over all queries:

$$\text{MAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\text{rank}_i} = \text{MRR}.$$

Note that in a single-positive-per-query setting, MAP is equivalent to MRR.

A.6.3.4 Normalized Discounted Cumulative Gain (nDCG@K)

nDCG evaluates the quality of the top-K results, with a logarithmic discount based on position. The Discounted Cumulative Gain (DCG) at rank K for query i is:

$$\text{DCG}_i\text{@K} = \sum_{k=1}^{K} \frac{\text{rel}_i(k)}{\log_2(k+1)}.$$

Since we have only one relevant document, $\text{rel}_i(k)$ is 1 only if the document at rank k is p_i^*. The Ideal DCG (IDCG) is the DCG score for a perfect ranking, where the relevant document is at rank 1. In our case, IDCG_i@K is 1 if K $\geq$ 1. The nDCG is the ratio of DCG to IDCG, averaged over all queries:

$$\text{nDCG@K} = \frac{1}{N}\sum_{i=1}^{N} \frac{\text{DCG}_i\text{@K}}{\text{IDCG}_i\text{@K}}.$$

This simplifies to $\frac{1}{\log_2(\text{rank}_i+1)}$ if $\text{rank}_i \leq K$, and 0 otherwise.

A.7 General-Domain Evaluation: SciFact

To assess whether AMLM-Fusion causes catastrophic forgetting of general linguistic knowledge, we evaluate both AMLM-Fusion and uniform BERT-MLM-CP on SciFact [231], a standard retrieval benchmark for scientific claim verification[1]. Table A.1 reports Recall@K and MRR for claim-to-passage retrieval under the same evaluation protocol used for financial QA matching (L2-normalized embeddings, cosine similarity).

Table A.1: AMLM-Fusion vs. BERT-MLM-CP on SciFact (allenai/scifact). Same protocol as financial QA matching. Higher is better.

Model	**Recall@1** ↑	**Recall@5** ↑	**Recall@10** ↑	**MRR** ↑
BERT-MLM-CP	0.5191	0.6728	0.7337	0.5892
AMLM-Fusion	0.4969	0.6699	0.6989	0.5371

[1] https://huggingface.co/datasets/allenai/scifact

AMLM-Fusion achieves significant improvements over BERT-MLM-CP on financial data (e.g., TheGoldmanEncyclopedia, Investopedia, and SmoothNLPNews), while remaining comparable to BERT-MLM-CP on standard retrieval benchmarks such as SciFact. This demonstrates that AMLM does not harm general linguistic performance and does not exhibit catastrophic forgetting when domain-adapted pre-training is applied with glossary-guided weighting.

A.8 External, Non-Comparable Baselines

This section reports off-the-shelf financial language models as reference points only: ProsusAI/finbert[2] and SALT-NLP/FLANG-BERT[3]. These models were *not* continued pre-trained on the corpus used in this work, whereas the main comparisons include models that were trained under identical continued pre-training protocols on the same data. To preserve fairness and internal validity, these baselines are excluded from the main tables and no performance claims are drawn against them. Tables A.2, A.3, and A.4 are included here strictly as rough external references under the exact same evaluation procedure used throughout this work.

Table A.2: Semantic similarity results for off-the-shelf models on the Investopedia set (N=22,940 pairs). Same protocol as main text. 95% CIs via Fisher (Pearson) and bootstrap (Spearman, Kendall).

Model	**Pearson [95% CI]** $\uparrow$	**Spearman [95% CI]** $\uparrow$	**Kendall [95% CI]** $\uparrow$	**MSE** $\downarrow$	**MAE** $\downarrow$
`FinBERT`	0.643 [0.635, 0.650]	0.601 [0.592, 0.610]	0.425 [0.418, 0.432]	0.013	0.090
`FLANG-BERT`	0.731 [0.725, 0.737]	0.666 [0.658, 0.674]	0.480 [0.473, 0.486]	0.010	0.078

Table A.3: Investopedia QA matching (N=22,940 pairs). Same protocol as main text.

Model	**Recall@1** $\uparrow$	**Recall@5** $\uparrow$	**Recall@10** $\uparrow$	**MRR** $\uparrow$	**nDCG@10** $\uparrow$
`FinBERT`	0.115 [0.111, 0.120]	0.276 [0.270, 0.281]	0.321 [0.315, 0.327]	0.195 [0.191, 0.199]	0.220 [0.216, 0.225]
`FLANG-BERT`	0.168 [0.163, 0.172]	0.384 [0.378, 0.390]	0.434 [0.428, 0.440]	0.271 [0.267, 0.276]	0.306 [0.301, 0.311]

Table A.4: TheGoldmanEncyclopedia QA matching (N=1,514 pairs). Same protocol as main text.

Model	**Recall@1** $\uparrow$	**Recall@5** $\uparrow$	**Recall@10** $\uparrow$	**MRR** $\uparrow$
`FinBERT`	0.026 [0.018, 0.034]	0.083 [0.070, 0.097]	0.127 [0.110, 0.144]	0.061 [0.053, 0.071]
`FLANG-BERT`	0.015 [0.009, 0.020]	0.070 [0.057, 0.083]	0.107 [0.092, 0.122]	0.049 [0.042, 0.057]

Higher correlations and R^2, and lower MSE/MAE, indicate better alignment with the reference.

[2] `https://huggingface.co/ProsusAI/finbert`
[3] `https://huggingface.co/SALT-NLP/FLANG-BERT`

A.9 Computational Resources and Training Time

All experiments were conducted on an `AWS EC2 'g5.2xlarge'`[4] instance, which is equipped with a single NVIDIA A10G Tensor Core GPU with 24 GB of memory. Training utilized mixed precision (FP16) to optimize memory usage and computational efficiency. The total training time per model was approximately 34-37 hours. The software stack used:

- **PyTorch** (`torch`) 2.7.1+cu128 (BSD-3-Clause)
- **Transformers** 4.54.0 (Apache-2.0)

A.10 Reproducibility Notes

- **Training Configuration**: Training uses `seed=42` via `transformers.set_seed`[5], which seeds Python's `random`, NumPy, and PyTorch RNGs (CPU/CUDA). We enable deterministic cuDNN (`deterministic=True`, `benchmark=False`)[6] and pass the same seed to DataLoader worker initialization to ensure consistent shuffling.
- **Evaluation Details**: The QA matching evaluation uses cosine similarity over L2- normalized embeddings with exact search. Intrinsic dimensionality estimation uses kNN-MLE on normalized embeddings. Cluster validity metrics use k-means clustering (with k set to the number of ground-truth topics) on normalized embeddings.
- **Stochastic Evaluation**: For stochastic procedures, evaluation metrics are averaged over three independent runs with different seeds (e.g., 42, 43, 44). We seed NumPy/Python accordingly for each run.
- **Limitations**: Multi-seed training repeats (reporting mean $\pm$ std) are left to future work due to compute constraints.

[4] https://aws.amazon.com/ec2/instance-types/g5/

[5] The `set_seed` utility in the Hugging Face Transformers library is used to set the random seeds for PyTorch, NumPy, and Python's standard `random` module, ensuring consistent initialization.

[6] These flags are set in PyTorch to configure the cuDNN backend. Setting `deterministic=True` ensures that cuDNN uses deterministic algorithms, and `benchmark=False` prevents it from choosing faster, non-deterministic algorithms. PyTorch reproducibility: https://docs.pytorch.org/docs/stable/notes/randomness.html.

Appendix B

CIR3

B.1 Metrics

B.1.1 Automatic Metrics

We provide a brief description of the metrics used in this study:

ROUGE-L [223] assesses recall by evaluating the overlap between reference and generated sentences using Longest Common Subsequence statistics. We use the implementation from GOOGLE[1]. We report the F1 score, the harmonic mean of precision and recall.

METEOR [224] is a recall-oriented metric that measures the similarity between generated and reference text, incorporating synonyms, stemming, and paraphrasing. We use the implementation from NLTK[2].

Jaccard Index[3][225] is a measure of similarity between two sets. It is calculated as the size of their intersection (elements they share) divided by the size of their union (total unique elements). Values range from 0 (no similarity) to 1 (identical sets). We adopt SCIKIT-LEARN's implementation[4].

[1]https://pypi.org/project/rouge-score
[2]https://www.nltk.org
[3]https://en.wikipedia.org/wiki/Jaccard_index
[4]https://scikit-learn.org

BERTScore[5] [163] uses contextual embeddings to assess word-level similarity via cosine similarity, correlating with human judgment in sentence and system evaluation, and providing precision, recall, and F1 metrics.

BAAI/bge-large[6] is a high-performance sentence embedding model, designed for semantic similarity tasks. It encodes text into dense vectors, allowing similarity to be measured via cosine similarity between embeddings.

B.1.2 Score Calculations

We denote $s(c, \mathcal{Q})$, $s(c, \mathcal{A})$, $s(\mathcal{Q}, \mathcal{A})$, $s(c, \mathcal{Q}_\oplus)$, $s(c, \mathcal{A}_\oplus)$, and $s(\mathcal{Q}_\oplus, \mathcal{A}_\oplus)$ the scores between (context and questions), (context and answers), (questions and answers), (context and concatenated questions), (context and concatenated answers), and (concatenated questions and concatenated answers), respectively. The scores are calculated as follows:

$$s(c, \mathcal{Q}) = \frac{1}{N} \sum_{i=1}^{N} s(c, q_i) \tag{B.1}$$

$$s(c, \mathcal{A}) = \frac{1}{N} \sum_{i=1}^{N} s(c, a_i) \tag{B.2}$$

$$s(\mathcal{Q}, \mathcal{A}) = \frac{1}{N} \sum_{i=1}^{N} s(q_i, a_i) \tag{B.3}$$

$$s(c, \mathcal{Q}_\oplus) = s(c, \oplus_{i=1}^{N} q_i) \tag{B.4}$$

$$s(c, \mathcal{A}_\oplus) = s(c, \oplus_{i=1}^{N} a_i) \tag{B.5}$$

$$s(\mathcal{Q}_\oplus, \mathcal{A}_\oplus) = s(\oplus_{i=1}^{N} q_i, \oplus_{i=1}^{N} a_i) \tag{B.6}$$

where s is the scoring function and $\oplus$ is the concatenation function.

B.2 CIR3: Algorithm Implementation Details

B.2.1 Module Input/Output Specifications

- **classify_subtopics**$(c : \text{str}, M : \text{int}) \rightarrow P : \text{List[str]}$

  Identifies subtopics within context.

[5] https://github.com/Tiiiger/bert_score
[6] https://github.com/FlagOpen/FlagEmbedding

- **generate_QAs**(c : str, M_{prev} : InnerMemory, H_{prev} : OuterMemory) $\rightarrow$ $\mathcal{G}^+$: List[QAPair]
  Generates QA pairs from context and memory.

- **refine_QAs**($\mathcal{G}^+$: List[QAPair], $\mathcal{M}$: InnerMemory) $\rightarrow \mathcal{F}$: Feedback $\cup$ $\{\varnothing\}$
  Produces refinement feedback or $\varnothing$ for consensus.

- **curmudgeon_QAs**($\mathcal{G}^-$: List[QAPair], $\mathcal{H}$: OuterMemory) $\rightarrow \mathcal{F}'$: Feedback $\cup$ $\{\varnothing\}$
  External evaluation feedback or $\varnothing$ for acceptance.

B.2.2 Error Handling

- **Subtopic identification failure**: Fallback to domain-specific default writer W_{p_0}.
- **Agent timeout**: 30-second timeout with retry mechanism (max 3 attempts).
- **Memory overflow**: Truncate oldest entries when memory exceeds max input tokens.
- **API failures**: Exponential back-off with graceful degradation to available models.
  1. Retry with exponential back-off (to handle transient failures).
  2. If still failing after a threshold (e.g. 5 attempts), gracefully degrade by using an alternative model / service.

B.3 Human Evaluation Guidelines

This section represents a partial excerpt of the evaluation guidelines.

Partial Excerpt of the Human Evaluation Guidelines.

This document describes the instructions of human evaluation for AI-based question-answer generation task. The results will be made publicly available within 12 months of study completion.

By proceeding with this task, you agree that any resulting work product may be shared publicly.

You will be provided with two datasets, each containing a collection of text documents and corresponding question-answer pairs. Your task is to evaluate the question-answer pairs in terms of their accuracy and completeness in relation to the information presented in the associated text documents.

Instructions:

Familiarise yourself with the context: Carefully read the provided context to understand the topic and key information it contains.

Review the generated question-answer pairs: Examine each question-answer pair.

Evaluate each criterion: For each of the following criteria, rate the question-answer pair on a scale of **1** to **5** (5 being the highest), and provide a brief explanation for your rating:

- **Coverage**: Does the question-answer pair address the main ideas and important details in the context?
- **Depth:** Does the question prompt deeper understanding of the context, or is it superficial? Does the answer provide sufficient detail and explanation?
- **Accuracy:** Is the answer factually correct and complete based on the information in the context?
- **Coherence:** Does the question-answer pair flow logically? Does the question naturally lead to the answer, and do they together contribute to a better understanding of the topic?
- **Representation:** Does the question-answer pair distort or present misleadingly any facts in the context?
- **Diversification:** Does the collection of question-answer pairs provide diverse and unique insights, or is there significant overlap in the knowledge they convey?

B.4 Evaluation Prompts

Following figures show the illustrative prompts used in our evaluation.

We also release CIR3's source code on GitHub[7].

```
"""
Create an Enhanced G-EVAL metric for Comprehensiveness evaluation based on Coverage, Depth, Accuracy, and Coherence aspects.
"""
EnhancedGEval(
    name="Comprehensiveness",
    evaluation_aspects=["Coverage", "Depth", "Accuracy", "Coherence"],
    evaluation_steps=[
        # Coverage
        "Examine the source document to identify all key topics, concepts, and important information covered",
        "Review the set of question-answer pairs to determine what aspects of the document they address",
        "Check if the questions cover all major themes and subtopics from the document",

        # Depth
        "Consider depth of coverage - are complex topics explored adequately or only superficially?",
        "Assess whether important details, relationships, and nuances are captured in the QA pairs",

        # Accuracy
        "Verify that the questions accurately reflect the document's content and answers are factually correct",

        # Coherence
        "Evaluate the logical flow and connection between questions and their relationship to document structure",

        # Final Assessment
        "Score HIGH if the QA set demonstrates comprehensive coverage, adequate depth, accuracy, and coherence",
```

[7]https://github.com/anonym-nlp-ai/cirrr

```
            "Score LOW if major topics are missing, coverage is superficial,
inaccurate, or lacks coherence",
        ],
        evaluation_params=[LLMTestCaseParams.INPUT,
LLMTestCaseParams.ACTUAL_OUTPUT],
        threshold=0.6,
        model="gpt-4o",
        top_logprobs=20,
        async_mode=True,
        verbose_mode=False,
        _include_g_eval_suffix=True
    )
```

Listing B.1: Comprehensiveness metric with four evaluation aspects: Coverage, Depth, Accuracy, and Coherence.

```
    """
    Create an Enhanced G-EVAL metric for Faithfulness evaluation based on
Accuracy, Exaggeration, Consistency, Justification, Plausibility, and
Misrepresentation aspects.
    """
    EnhancedGEval(
        name="Faithfulness",
        evaluation_aspects=[
            "Accuracy", "Exaggeration", "Consistency",
            "Justification", "Plausibility", "Misrepresentation"
        ],
        evaluation_steps=[
            # Accuracy
            "Carefully read the source document to understand the factual
information presented",
            "Examine each answer to verify factual accuracy against the source
document",

            # Exaggeration
            "Check for any statements that overstate or embellish information
from the document",

            # Consistency
            "Look for contradictions or deviations from facts presented in the
source material",
```

```

        # Justification
        "Verify that all claims in answers are well-supported by evidence from the document",

        # Plausibility
        "Assess whether answers represent reasonable inferences based on the document content",

        # Misrepresentation
        "Check for any distortion or misleading presentation of facts from the source",

        # Final Assessment
        "Score HIGH if answers demonstrate accuracy, avoid exaggeration, maintain consistency, are well-justified, plausible, and avoid misrepresentation",
        "Score LOW if answers contain inaccuracies, exaggerations, inconsistencies, poor justification, implausible claims, or misrepresentations",
    ],
    evaluation_params=[LLMTestCaseParams.INPUT, LLMTestCaseParams.ACTUAL_OUTPUT],
    threshold=0.6,
    model="gpt-4o",
    top_logprobs=20,
    async_mode=True,
    verbose_mode=False,
    _include_g_eval_suffix=True
)
```

Listing B.2: Faithfulness metric with six evaluation aspects: Accuracy, Exaggeration, Consistency, Justification, Plausibility, and Misrepresentation.

B.5 Case Study: QA Evolution Trajectory Analysis

To demonstrate how iterative feedback tangibly improves QA diversity and alignment, we present a detailed trajectory analysis of a representative financial document undergoing CIR3's refinement process.

Multi-Iteration Refinement Example

Context: "A strong financial plan balances growth, stability, and protection by strategically combining diverse assets. Bonds provide steady income and preserve capital, though they are subject to inflation and interest rate risks. For growth, index ETFs offer low-cost, diversified exposure to the stock market, accepting the associated market volatility. A structured retirement is built on a foundation of pensions, while annuities provide guaranteed lifetime income, guarding against the possibility of outliving one's savings. Finally, life insurance protects dependents and can build cash value for future needs. By aligning these tools with one's personal risk tolerance, time horizon, and specific goals, a resilient financial strategy can be built to adapt to economic changes and secure both a current lifestyle and future legacy."

Identified Subtopics:

- financial planning
- retirement
- investment
- risk management

Writer Agent Assignment:

- $W_{\texttt{financial planning}}$: General financial strategy perspective;
- $W_{\texttt{retirement}}$: Retirement-focused analysis;
- $W_{\texttt{investment}}$: Investment and growth perspective;
- $W_{\texttt{risk management}}$: Risk assessment and mitigation;

// ...

Outer Iteration k=1

Inner Iteration l=4 - Writers Collaboration: QA generation:

- Q: What is the main objective of a strong financial plan?
  A: To balance growth, stability, and protection by strategically combining diverse assets.
- Q: Which investment provides steady income and preserves capital?
  A: Bonds provide steady income and preserve capital.
- Q: What is the main role of index ETFs in a financial plan?
  A: They offer low-cost, diversified exposure to the stock market.
- Q: What forms the foundation of a structured retirement?
  A: Pensions form the foundation of a structured retirement.
- Q: What type of investment offers guaranteed lifetime income?
  A: Annuities offer guaranteed lifetime income.

Vendi Diversity Scores

- "balanced_g_score": 0.6598721146583557,
- "score_a": 1.3656857,
- "score_ca": 1.0813369,
- "score_q": 1.4364766

Curmudgeon reflection: The current QA set provides a good foundation, covering key topics such as the main objective of a financial plan, the roles of bonds, index ETFs, pensions, and annuities. However, it falls short in exploring the nuances of aligning financial strategies with personal circumstances and misses the role of life insurance. The answers accurately reflect the information in the document, but the question set lacks diversity, primarily focusing on factual recall with similar structures and complexity. To improve, it's essential to add more analytical and comparative questions that require deeper analysis and synthesis, as well as varying the linguistic patterns and cognitive demands.
Curmudgeon status: "refine"
// ...

Outer Iteration k=2
// ...

Inner Refinement Feedback (l=2): $W_{\text{retirement}}$: "Missing coverage of annuities and longevity risk protection." $W_{\text{risk management}}$: "Need specific risk details: inflation, interest rate, market volatility impacts." $W_{\text{financial planning}}$: "Lacks personalization factors: risk tolerance, time horizon alignment." $W_{\text{investment}}$: "Should clarify ETF advantages beyond basic market exposure."

Inner Iteration l=2 - Writers Collaboration: QA generation:

- Q: How does a strong financial plan integrate different asset types?
  A: It combines growth assets like index ETFs, stability assets like bonds, and protection assets like pensions, annuities, and life insurance.
- Q: What risks can affect the income from bonds?
  A: Bonds are subject to inflation and interest rate risks.
- Q: What trade-off comes with investing in index ETFs?
  A: They provide market growth potential at low cost but come with market volatility.
- Q: How do annuities support retirement security?
  A: They provide guaranteed lifetime income, guarding against the possibility of outliving one's savings.
- Q: What is the role of life insurance in financial protection?
  A: It protects dependents and can build cash value for future needs.
- Q: How can financial tools be matched to personal needs?
  A: By aligning them with personal risk tolerance, time horizon, and specific goals.

- Q: What is a key advantage of bonds compared to index ETFs?
  A: Bonds guarantee higher returns over time with minimum risk of loss, making them a safer alternative for long-term investors seeking consistent growth.

Vendi Diversity Scores

- "balanced_g_score": 0.7028632760047913,
- "score_a": 1.6247656,
- "score_ca": 1.1747185,
- "score_q": 1.5361243

Curmudgeon reflection: The QA set are mostly aligned with the document and covers most major themes from the document, including the integration of different asset types, risks associated with bonds, trade-offs of index ETFs, and roles of annuities and life insurance. It also addresses aligning financial tools with personal needs. However, Q7 contains a factual inaccuracy, as the document does not state that bonds guarantee higher returns over time or that they are a safer alternative for consistent growth. This misrepresentation reduces the depth of coverage and overall alignment with the source material. To further refine, it's crucial to ensure the accuracy of all questions and maintain a diverse range of question types and complexities.
Curmudgeon status: "refine"

Inner Iteration l=2 - Writers Collaboration: QA generation:

- Q: How does strategic asset allocation create a resilient financial plan?
  A: By balancing growth through index ETFs, stability through bonds, and protection through pensions, annuities, and life insurance.
- Q: In what way do bonds contribute to stability but face risks?
  A: They provide steady income and preserve capital but are affected by inflation and interest rate changes.
- Q: How do index ETFs contribute to long-term growth?
  A: They provide diversified stock market exposure at low cost while accepting market volatility.
- Q: Why are pensions important in retirement planning?
  A: They provide a stable foundation for a structured retirement.
- Q: What dual benefit do annuities offer in retirement?
  A: They ensure guaranteed lifetime income and protect against outliving one's savings.
- Q: How does life insurance support both protection and future planning?
  A: It safeguards dependents and can accumulate cash value for future needs.
- Q: What personal factors shape an adaptive financial strategy?
  A: Risk tolerance, time horizon, and specific goals.

```
• Q: How can a financial plan remain effective during economic changes?
  A: By aligning growth, stability, and protection tools with personal needs.

Vendi Diversity Scores

• "balanced_g_score": 0.741766095161438
• "score_a": 1.5003632
• "score_ca": 1.0616497
• "score_q": 1.5900005

Curmudgeon reflection: ""
Curmudgeon: "agreement"
Curmudgeon agreement feedback: The QA set has achieved a high level of quality,
comprehensively covering all major themes and subtopics from the document, including
strategic asset allocation, personal factors, and the roles of various financial
instruments. The answers align perfectly with the source document, maintaining the tone
and perspective without contradictions or inaccuracies. The question set demonstrates
a strong variety in types, including factual, analytical, and comparative questions,
and explores diverse aspects of financial planning from different angles. The set
effectively covers different levels of detail and maintains linguistic diversity, making
it ready for agreement without further refinement.
```

B.5.1 Qualitative Curmudgeon Feedback Analysis

The curmudgeon's iterative feedback (Listing: Multi-Iteration Refinement Example) reveals concrete improvement mechanisms:

- **Iteration 1**: Identifies specific coverage gaps (*"misses the role of life insurance"*) and structural limitations (*"question set lacks diversity, primarily focusing on factual recall"*). The feedback provides actionable guidance: *"add more analytical and comparative questions"* and *"varying the linguistic patterns and cognitive demands."*.
- **Iteration 2**: Demonstrates error detection capabilities by identifying a critical factual inaccuracy in Q7 (*"the document does not state that bonds guarantee higher returns over time"*). This shows the curmudgeon's ability to catch hallucination errors that could compromise faithfulness.
- **Iteration 3**: Confirms convergence with comprehensive validation: *"covers all major themes and subtopics"*, *"answers align perfectly with the source document"* and *"demonstrates strong variety in types, including factual, analytical, and comparative questions."*

B.5.1.1 Common Error Mitigation Evidence

The trajectory demonstrates CIR3's effectiveness against the four primary error categories:

- **Duplication**: The increasing Vendi Score for questions (1.436 → 1.590) indicates successful reduction of redundant or overly similar questions. The curmudgeon's observation about *"similar structures and complexity"* in Iteration 1 implicitly addresses this concern.
- **Hallucination**: Explicitly detected by the curmudgeon in Iteration 2, identifying Q7's factual inaccuracy (*"the document does not state that bonds guarantee higher returns over time"*) as hallucinated information not present in the source document.
- **Irrelevance**: The curmudgeon's focus on coverage gaps (*"misses the role of life insurance"*) and requirement for document alignment (*"answers align perfectly with the source document"*) addresses relevance, even if this error type is not explicitly flagged in this particular trajectory.
- **Specificity**: Addressed through the curmudgeon's feedback requesting *"varying the linguistic patterns and cognitive demands"* and *"more analytical and comparative questions"* which counters template driven or overly narrow question generation.

B.5.2 Quantitative Evolution Patterns

Tables B.2 and B.1 reveal a systematic improvement in key metrics with notable intermediate dynamics:

Table B.1: Vendi Score diversity metrics for questions, answers, context-answers, and balanced G score across iterations.

		Vendi Scores			
Iteration	**QA Count**	**Questions**	**Answers**	**Context-Answers**	**Balanced G**
#1	5	1.436	1.366	1.081	0.660
#2	7	1.536	1.625	1.175	0.703
#3	8	1.590	1.500	1.062	0.742

Table B.2: Comprehensiveness and Faithfulness scores across iterations for QA trajectory.

Iteration	QA Count	Comprehensiveness	Faithfulness
#1	5	0.734	1.000
#2	7	0.803	0.767
#3	8	0.953	0.999

- **Comprehensiveness** shows substantial growth from 0.734 (Iteration 1) to 0.953 (Iteration 3), representing a 29.85% improvement. QA Count expands strategically from 5 to 8 questions, indicating a controlled expansion of coverage rather than arbitrary multiplication.

- **Faithfulness** demonstrates an interesting trajectory: starting at perfect alignment (1.00), it drops to 0.767 in Iteration 2 as new content is added, then recovers to near-perfect alignment (0.999) in Iteration 3. Similarly, Vendi Score A (answer diversity) shows a comparable pattern, peaking at 1.625 in Iteration 2 before stabilizing at 1.5 in Iteration 3. These temporary fluctuations reflect the natural trade-off between coverage expansion and maintaining document fidelity, a pattern that validates the curmudgeon's role in balancing comprehensiveness, diversity, and faithfulness through iterative refinement.

- **Vendi Score** patterns demonstrate the curmudgeon's dual optimization: Question diversity increases consistently from 1.436 to 1.59, while the balanced generation score (G) improves from 0.66 to 0.742, showing enhanced overall system performance despite the intermediate adjustments in answer diversity.

B.6 Computational Cost Analysis

CIR3's multi-agent architecture with iterative refinement requires careful consideration of computational resources for practical deployment. We provide detailed cost breakdowns across different infrastructure configurations to enable informed feasibility assessments.

Table B.3: Average CIR3 computational resource utilization for 1000 documents, comparing Groq cloud API deployment and self-hosted p5.48xlarge instance with vLLM, showing token consumption and round-trip inference times per component.

Component	**Tokens**	**Round trip inference time (s)**	
		Groq	**AWS p5.48xlarge + vLLM**
Classifier Agent	25	0.2	0.26
Moderator Agent	2007	1.7	2.17
Writer Agents	17287	13.71	17.54
Curmudgeon Agent	617	11.28	14.43
Diversity (Encoder)	-	6.55	6.55
Total	**19937**	**38**	**48.51**

Table B.3 presents comprehensive resource utilization metrics for CIR3 across two deployment scenarios: cloud-based API services (Groq[8]) and self-hosted infrastructure (p5.48xlarge[9] with

[8] https://groq.com/pricing
[9] https://instances.vantage.sh/aws/ec2/p5.48xlarge?currency=USD

vLLM). The total token consumption per document averages 19937 tokens across all agents, with writer agents consuming the majority (17287 tokens, 86.7%) due to their iterative QA generation and refinement processes.

B.6.1 Component-Level Analysis

The computational distribution reveals strategic resource allocation: the classifier agent requires minimal resources (25 tokens, 0.2-0.26s), enabling efficient subtopic identification. The moderator agent consumes 2007 tokens (1.7-2.17s) for coordination tasks, while the diversity encoder requires 6.5 seconds for Vendi Score computation via a self-hosted embeddings API[10]. The curmudgeon agent runs in 11.28–14.43 seconds across configurations, reflecting its evaluation and feedback generation steps.

B.6.2 Infrastructure Performance Analysis

To provide comprehensive feasibility assessment, we evaluated CIR3 across two distinct deployment architectures with detailed performance characteristics.

Table B.4: Comparative performance analysis of CIR3 deployment configurations showing throughput, latency characteristics, and operational trade-offs.

Metric	**Groq (LPU)**	**AWS p5.48xlarge + vLLM**
Throughput (tokens/sec)	525	411
Total Processing Time (s)	38.0	48.5
Time to First Token (s)	0.22-0.3	0.2-0.4
Latency Consistency	Deterministic	Variable
Operational Complexity	Minimal	High
Deployment Flexibility	Limited	Full Control

Groq Cloud Deployment: Groq's Language Processing Unit (LPU) architecture delivered consistent performance, processing 19937 tokens in 38.0 seconds with sustained throughput of approximately 525 tokens per second (Table B.4). The deterministic latency characteristics (time-to-first-token: 0.22-0.3s) prove particularly valuable for multi-agent systems where round-trip delays compound across iterative cycles.

Self-Hosted AWS Configuration: Our optimized AWS p5.48xlarge deployment (8xH100 GPUs with vLLM) achieved comparable performance: 19937 tokens in 48.5 seconds, sustaining approximately 411 tokens per second. Key optimizations included continuous batching for

[10] `https://github.com/huggingface/text-embeddings-inference`

pipeline utilization, KV cache management to prevent memory thrashing, CUDA graphs with FlashAttention for latency reduction, and tuned EFA/NCCL communication across GPUs[11].

B.6.3 Practical Implications

The performance differential between configurations is modest (22% throughput difference), with Groq achieving superior consistency while our AWS setup provides greater customization capabilities. Both configurations demonstrate CIR3's practical feasibility for production deployment. The throughput rates (411-525 tokens/second) support real-time document processing applications, while the total processing time (38-49 seconds per document) remains reasonable for comprehensive QA generation tasks. Despite the multi-agent complexity, CIR3 demonstrates favorable cost-benefit ratios when considering the substantial quality improvements. The modular architecture enables selective deployment optimization, such as caching classifier results or parallelizing writer agent operations, making the framework adaptable to various computational budget constraints.

B.6.3.1 Cost Analysis for Knowledge Base Processing

To evaluate CIR3's economic feasibility for large-scale deployment, we analyze the costs of processing a knowledge base of 1000 documents across multiple infrastructure configurations.

Table B.5: Cost analysis for processing 1000 documents across deployment configurations. **Note**: Cost and time estimates are based on sequential execution and do not account for batch processing optimizations, which would reduce these numbers considerably.

Configuration	Total Cost	Cost per Document	Processing Time
Groq Cloud (Llama 3 70B)	$16.90	$0.0169	10.6 hours
AWS p5.48xlarge (8xH100, Spot)	$253.07	$0.25	13.5 hours
AWS p4de.24xlarge (8xA100, Spot)	$158.65	$0.16	13.5 hours*

*Estimated processing time; actual performance may vary.

Groq Cloud Pricing: Based on Groq's current pricing structure [232], using Llama 3 70B (our primary model) costs $0.59 per million input tokens and $0.79 per million output tokens. For CIR3's token distribution (19,937 tokens per document with approximately 70% input and 30% output), the cost per document is approximately $0.0169. Processing 1000 documents

[11]Elastic Fabric Adapter (EFA) enables low-latency interconnect on AWS; NCCL provides high-performance multi-GPU/multi-node collectives: https://developer.nvidia.com/nccl https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/efa.html

would cost approximately **$16.90** in token fees, with processing completing in approximately 10.6 hours (Table B.5).

AWS Self-Hosted Configurations: We evaluated two AWS configurations for self-hosted deployment:

- **p5.48xlarge (8xH100):** At $18.746 per hour spot pricing[12], processing 1000 documents (13.5 hours) costs approximately $253.07
- **p4de.24xlarge (8xA100):** At $11.752 per hour spot pricing, the same workload costs approximately $158.65

It is important to note that in practical deployments, self-hosted infrastructure typically serves multiple applications beyond CIR3, effectively amortizing costs across various workloads. This shared utilization significantly improves the cost-effectiveness of dedicated GPU clusters for organizations running multiple AI applications.

B.6.3.2 Cost-Benefit Analysis

The substantial quality improvements achieved by CIR3 (+23% comprehensiveness, +17% faithfulness compared to baselines) justify the computational overhead for applications requiring high-quality QA generation. CIR3 is particularly well-suited for batch processing scenarios such as:

- Knowledge base creation and augmentation for enterprise documentation.
- Dataset indexing and enrichment for information retrieval systems.
- Educational content processing for automated quiz and assessment generation.
- Research literature analysis for systematic review and knowledge extraction.
- Financial document processing for banking, investment analysis, and regulatory compliance.
- Legal and compliance document processing for searchable QA databases.
- Scheduled document processing pipelines for content management systems.

[12] AWS pricing may vary across regions: https://aws.amazon.com/ec2/spot/pricing

For batch processing workflows, AWS spot pricing offers significant cost advantages, with the p4de.24xlarge configuration providing a favorable balance between performance and cost. When infrastructure is shared across multiple applications, the effective cost per CIR3 document decreases substantially, making large-scale deployment highly economical.

Bibliography

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," July 2020.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[3] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 8342–8360, Association for Computational Linguistics, July 2020.

[4] P. Atanasova, O.-M. Camburu, C. Lioma, T. Lukasiewicz, J. G. Simonsen, and I. Augenstein, "Faithfulness Tests for Natural Language Explanations," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 283–294, Association for Computational Linguistics, July 2023.

[5] P. Schmidtova, S. Mahamood, S. Balloccu, O. Dusek, A. Gatt, D. Gkatzia, D. M. Howcroft, O. Platek, and A. Sivaprasad, "Automatic Metrics in Natural Language Generation: A survey of Current Evaluation Practices," in *Proceedings of the 17th International Natural*

Language Generation Conference (S. Mahamood, N. L. Minh, and D. Ippolito, eds.), (Tokyo, Japan), pp. 557–583, Association for Computational Linguistics, Sept. 2024.

[6] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, pp. 1234–1240, Feb. 2020.

[7] D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models," Aug. 2019. arXiv:1908.10063 [cs].

[8] L. Zhang, Q. Chen, W. Wang, C. Deng, X. Cao, K. Hao, Y. Jiang, and W. Wang, "Weighted Sampling for Masked Language Modeling," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, June 2023. ISSN: 2379-190X.

[9] D. Yang, Z. Zhang, and H. Zhao, "Learning Better Masking for Better Language Model Pre-training," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 7255–7267, Association for Computational Linguistics, July 2023.

[10] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving Factuality and Reasoning in Language Models through Multiagent Debate," May 2023.

[11] T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, and S. Shi, "Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate," May 2023.

[12] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," Mar. 2023.

[13] D. Deutsch, R. Dror, and D. Roth, "On the Limitations of Reference-Free Evaluations of Generated Text," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (Y. Goldberg, Z. Kozareva, and Y. Zhang, eds.), (Abu Dhabi, United Arab Emirates), pp. 10960–10977, Association for Computational Linguistics, Dec. 2022.

[14] T. Ito, K. v. Deemter, and J. Suzuki, "Reference-free Evaluation Metrics for Text Generation: A Survey," Jan. 2025. arXiv:2501.12011 [cs].

[15] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," Dec. 2023. arXiv:2306.05685 [cs].

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), pp. 6000–6010, Curran Associates Inc., Dec. 2017.

[17] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.

[18] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179–211, Apr. 1990.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep Contextualized Word Representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (M. Walker, H. Ji, and A. Stent, eds.), (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[20] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI*, 2019. Accessed: 2024-11-15.

[21] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Sept. 2023.

[22] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 7871–7880, Association for Computational Linguistics, July 2020.

[23] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," Feb. 2020. arXiv:1909.11942 [cs].

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," July 2019. arXiv:1907.11692 [cs].

[25] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving Pre-training by Representing and Predicting Spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020. Place: Cambridge, MA Publisher: MIT Press.

[26] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," ,, 2018.

[27] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546 vol. 1, June 2005. ISSN: 1063-6919.

[28] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, eds.), (Online and Punta Cana, Dominican Republic), pp. 6894–6910, Association for Computational Linguistics, Nov. 2021.

[29] H. Hu, X. Wang, Y. Zhang, Q. Chen, and Q. Guan, "A comprehensive survey on contrastive learning," *Neurocomputing*, vol. 610, p. 128645, Dec. 2024.

[30] E. Alhazmi, Q. Z. Sheng, W. E. Zhang, M. I. Thanoon, H. Zhuang, B. Soltani, and M. Zaib, "Fine-Tuning Encoder-Decoder Models with Contrastive Learning for In-Context Distractor Generation," in *Findings of the Association for Computational Linguistics: EMNLP 2025* (C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, eds.), (Suzhou, China), pp. 10056–10072, Association for Computational Linguistics, Nov. 2025.

[31] G. et al, "The Llama 3 Herd of Models," Nov. 2024. arXiv:2407.21783 [cs].

[32] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan,

T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, and Z. Fan, "Qwen2 Technical Report," Sept. 2024. arXiv:2407.10671 [cs].

[33] Qwen, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 Technical Report," Jan. 2025. arXiv:2412.15115 [cs].

[34] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-VL Technical Report," Feb. 2025. arXiv:2502.13923 [cs].

[35] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, "Qwen3 Technical Report," May 2025. arXiv:2505.09388 [cs].

[36] A. et al, "Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone," Aug. 2024. arXiv:2404.14219 [cs].

[37] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, J. R. Lee, Y. T. Lee, Y. Li, W. Liu, C. C. T. Mendes, A. Nguyen, E. Price, G. d. Rosa, O. Saarikivi, A. Salim, S. Shah, X. Wang, R. Ward, Y. Wu, D. Yu, C. Zhang, and Y. Zhang, "Phi-4 Technical Report," Dec. 2024. arXiv:2412.08905 [cs].

[38] T. et al, "Gemma 3 Technical Report," Mar. 2025. arXiv:2503.19786 [cs].

[39] Anthropic, "Claude 4 Model Overview," 2025. Whitepaper.

[40] OpenAI and Hurst, "GPT-4o System Card," Oct. 2024. arXiv:2410.21276 [cs].

[41] G. e. a. Comanici, “Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities,” Oct. 2025. arXiv:2507.06261 [cs].

[42] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, and Y. Koreeda, “Holistic Evaluation of Language Models,” Oct. 2023. arXiv:2211.09110 [cs].

[43] S. CRFM, “Helm: Holistic Evaluation of Language Models (repository),” 2023.

[44] L. Zheng, W.-L. Chiang, Y. Sheng, T. Li, S. Zhuang, Z. Wu, Y. Zhuang, Z. Li, Z. Lin, E. P. Xing, J. E. Gonzalez, I. Stoica, and H. Zhang, “LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset,” Mar. 2024. arXiv:2309.11998 [cs].

[45] LMSYS, “Lmsys Chatbot Arena Leaderboard,” 2024.

[46] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, “Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation,” Oct. 2023. arXiv:2305.01210 [cs].

[47] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, “Evalplus (repository),” 2023.

[48] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer,” Jan. 2017. arXiv:1701.06538 [cs].

[49] X. Du, T. Gunter, X. Kong, M. Lee, Z. Wang, A. Zhang, N. Du, and R. Pang, “Revisiting MoE and Dense Speed-Accuracy Comparisons for LLM Training,” June 2024. arXiv:2405.15052 [cs].

[50] DeepSeek-AI, “DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model,” June 2024. arXiv:2405.04434 [cs].

[51] DeepSeek-AI, “DeepSeek-V2 (repository),” 2024.

[52] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (I. Gurevych and Y. Miyao, eds.), (Melbourne, Australia), pp. 2126–2136, Association for Computational Linguistics, July 2018.

[53] M. E. Peters, S. Ruder, and N. A. Smith, "To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)* (I. Augenstein, S. Gella, S. Ruder, K. Kann, B. Can, J. Welbl, A. Conneau, X. Ren, and M. Rei, eds.), (Florence, Italy), pp. 7–14, Association for Computational Linguistics, Aug. 2019.

[54] H. El Boukkouri, O. Ferret, T. Lavergne, and P. Zweigenbaum, "Re-train or Train from Scratch? Comparing Pre-training Strategies of BERT in the Medical Domain," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference* (N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, eds.), (Marseille, France), pp. 2626–2633, European Language Resources Association, June 2022.

[55] Y. Xie, K. Aggarwal, and A. Ahmad, "Efficient Continual Pre-training for Building Domain Specific Large Language Models," in *Findings of the Association for Computational Linguistics: ACL 2024* (L.-W. Ku, A. Martins, and V. Srikumar, eds.), (Bangkok, Thailand), pp. 10184–10201, Association for Computational Linguistics, Aug. 2024.

[56] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," in *Findings of the Association for Computational Linguistics: EMNLP 2020* (T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 2898–2904, Association for Computational Linguistics, Nov. 2020.

[57] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, pp. 128–135, Apr. 1999. Publisher: Elsevier.

[58] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, pp. 3521–3526, Mar. 2017. arXiv:1612.00796 [cs].

[59] A. Ramesh Kashyap, L. Mehnaz, B. Malik, A. Waheed, D. Hazarika, M.-Y. Kan, and R. R. Shah, "Analyzing the Domain Robustness of Pretrained Language Models, Layer by Layer," in *Proceedings of the Second Workshop on Domain Adaptation for NLP* (E. Ben-David, S. Cohen, R. McDonald, B. Plank, R. Reichart, G. Rotman, and Y. Ziser, eds.), (Kyiv, Ukraine), pp. 222–244, Association for Computational Linguistics, Apr. 2021.

[60] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (K. Inui, J. Jiang, V. Ng, and X. Wan, eds.), (Hong Kong, China), pp. 3615–3620, Association for Computational Linguistics, Nov. 2019.

[61] C. Liu, S. Wang, L. Qing, K. Kuang, Y. Kang, C. Sun, and F. Wu, "Gold Panning in Vocabulary: An Adaptive Method for Vocabulary Expansion of Domain-Specific LLMs," Oct. 2024. arXiv:2410.01188 [cs].

[62] V. S. Kohli, A. Monis, and R. Mamidi, "Choose Your Words Wisely: Domain-adaptive Masking Makes Language Models Learn Faster," in *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)* (V. Adlakha, A. Chronopoulou, X. L. Li, B. P. Majumder, F. Shi, and G. Vernikos, eds.), (Albuquerque, NM), pp. 87–91, Association for Computational Linguistics, May 2025.

[63] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced Language Representation with Informative Entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 1441–1451, Association for Computational Linguistics, July 2019.

[64] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. d. Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-Efficient Transfer Learning for NLP," June 2019. arXiv:1902.00751 [cs].

[65] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," Oct. 2021. arXiv:2106.09685 [cs].

[66] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (I. Gurevych and Y. Miyao, eds.), (Melbourne, Australia), pp. 328–339, Association for Computational Linguistics, July 2018.

[67] X. Zhang, P. Shapiro, G. Kumar, P. McNamee, M. Carpuat, and K. Duh, "Curriculum Learning for Domain Adaptation in Neural Machine Translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), (Minneapolis, Minnesota), pp. 1903–1915, Association for Computational Linguistics, June 2019.

[68] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," Feb. 2018. arXiv:1708.02002 [cs].

[69] C. Lee, K. Cho, and W. Kang, "Mixout: Effective Regularization to Finetune Large-scale Pretrained Language Models," Jan. 2020. arXiv:1909.11299 [cs].

[70] R. Müller, S. Kornblith, and G. Hinton, "When Does Label Smoothing Help?," June 2020. arXiv:1906.02629 [cs].

[71] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jan. 2014.

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015. arXiv:1512.03385 [cs].

[73] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," July 2016. arXiv:1607.06450 [stat].

[74] A. Ziaee and E. ÇAno, "Batch Layer Normalization A new normalization layer for CNNs and RNNs," in *Proceedings of the 6th International Conference on Advances in Artificial Intelligence*, ICAAI '22, (New York, NY, USA), pp. 40–49, Association for Computing Machinery, Jan. 2023.

[75] C. de Masson d' Autume, S. Ruder, L. Kong, and D. Yogatama, "Episodic Memory in Lifelong Language Learning," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[76] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), pp. 6470–6479, Curran Associates Inc., Dec. 2017.

[77] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Mar. 2015. arXiv:1503.02531 [stat].

[78] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive Neural Networks," Oct. 2016. arXiv:1606.04671 [cs].

[79] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," Sept. 2019. arXiv:1706.06083 [stat].

[80] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Dou, and J.-R. Wen, "Large Language Models for Information Retrieval: A Survey," Jan. 2024.

[81] L. Silva and L. Barbosa, "Improving dense retrieval models with LLM augmented data for dataset search," *Knowledge-Based Systems*, vol. 294, p. 111740, June 2024.

[82] D. Lindberg, F. Popowich, J. Nesbit, and P. Winne, "Generating Natural Language Questions to Support Learning On-Line," in *Proceedings of the 14th European Workshop on Natural Language Generation* (A. Gatt and H. Saggion, eds.), (Sofia, Bulgaria), pp. 105–114, Association for Computational Linguistics, Aug. 2013.

[83] N.-T. Le, T. Kojiri, and N. Pinkwart, "Automatic Question Generation for Educational Applications – The State of Art," in *Advanced Computational Methods for Knowledge Engineering* (T. van Do, H. A. L. Thi, and N. T. Nguyen, eds.), (Cham), pp. 325–338, Springer International Publishing, 2014.

[84] R. Puri, R. Spring, M. Shoeybi, M. Patwary, and B. Catanzaro, "Training Question Answering Models From Synthetic Data," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 5811–5826, Association for Computational Linguistics, Nov. 2020.

[85] M. Heilman and N. A. Smith, "Good Question! Statistical Ranking for Question Generation," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (R. Kaplan, J. Burstein,

M. Harper, and G. Penn, eds.), (Los Angeles, California), pp. 609–617, Association for Computational Linguistics, June 2010.

[86] J. Mostow and W. Chen, "Generating Instruction Automatically for the Reading Strategy of Self-Questioning," *International Conference on Artificial Intelligence in Education*, vol. -, no. -, 2009.

[87] X. Du, J. Shao, and C. Cardie, "Learning to Ask: Neural Question Generation for Reading Comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (R. Barzilay and M.-Y. Kan, eds.), (Vancouver, Canada), pp. 1342–1352, Association for Computational Linguistics, July 2017.

[88] M. Alaofi, L. Gallagher, M. Sanderson, F. Scholer, and P. Thomas, "Can Generative LLMs Create Query Variants for Test Collections? An Exploratory Study," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, (New York, NY, USA), pp. 1869–1873, Association for Computing Machinery, July 2023.

[89] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (J. Su, K. Duh, and X. Carreras, eds.), (Austin, Texas), pp. 2383–2392, Association for Computational Linguistics, Nov. 2016.

[90] D. Weissenborn, G. Wiese, and L. Seiffe, "Making Neural QA as Simple as Possible but not Simpler," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)* (R. Levy and L. Specia, eds.), (Vancouver, Canada), pp. 271–280, Association for Computational Linguistics, Aug. 2017.

[91] C. Alberti, D. Andor, E. Pitler, J. Devlin, and M. Collins, "Synthetic QA Corpora Generation with Roundtrip Consistency," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 6168–6173, Association for Computational Linguistics, July 2019.

[92] A. Ushio, F. Alva-Manchego, and J. Camacho-Collados, "An Empirical Comparison of LM-based Question and Answer Generation Methods," in *Findings of the Association for Computational Linguistics: ACL 2023* (A. Rogers, J. Boyd-Graber, and N. Okazaki,

eds.), (Toronto, Canada), pp. 14262–14272, Association for Computational Linguistics, July 2023.

[93] 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*, vol. 40, pp. 1–43, Jan. 2022.

[94] S. Vakulenko, B. Byrne, and A. de Gispert, "Uniform Training and Marginal Decoding for Multi-Reference Question-Answer Generation," in *ECAI 2023*, pp. 2378–2385, IOS Press, 2023.

[95] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in *The Twelfth International Conference on Learning Representations*, Oct. 2023.

[96] V. Puranik, A. Majumder, and V. Chaoji, "PROTEGE: Prompt-based Diverse Question Generation from Web Articles," in *Findings of the Association for Computational Linguistics: EMNLP 2023* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 5449–5463, Association for Computational Linguistics, Dec. 2023.

[97] W. Zhang, W. Hua, and K. Stratos, "EntQA: Entity Linking as Question Answering," Mar. 2022.

[98] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," Jan. 2020. arXiv:2001.08361 [cs].

[99] T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate Before Use: Improving Few-Shot Performance of Language Models," June 2021. arXiv:2102.09690 [cs].

[100] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao, and X. He, "Exploring Large Language Model based Intelligent Agents: Definitions, Methods, and Prospects," Jan. 2024.

[101] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," Jan. 2023.

[102] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large Language Models are Zero-Shot Reasoners," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22199–22213, Dec. 2022.

[103] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 2609–2634, Association for Computational Linguistics, July 2023.

[104] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," Dec. 2023.

[105] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, and T. Hoefler, "Graph of Thoughts: Solving Elaborate Problems with Large Language Models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 17682–17690, Mar. 2024.

[106] P. Zhou, J. Pujara, X. Ren, X. Chen, H.-T. Cheng, Q. V. Le, E. H. Chi, D. Zhou, S. Mishra, and H. S. Zheng, "Self-Discover: Large Language Models Self-Compose Reasoning Structures," Feb. 2024. arXiv:2402.03620 [cs].

[107] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," Mar. 2023. arXiv:2210.03629 [cs].

[108] B. Xu, Z. Peng, B. Lei, S. Mukherjee, Y. Liu, and D. Xu, "ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models," May 2023.

[109] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks," Oct. 2023. arXiv:2211.12588 [cs].

[110] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig, "PAL: Program-aided Language Models," Jan. 2023. arXiv:2211.10435 [cs].

[111] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language Models Can Teach Themselves to Use Tools," Feb. 2023. arXiv:2302.04761 [cs].

[112] S. Kim, S. Moon, R. Tabrizi, N. Lee, M. W. Mahoney, K. Keutzer, and A. Gholami, "An LLM Compiler for Parallel Function Calling," Feb. 2024.

[113] A. Zhou, K. Yan, M. Shlapentokh-Rothman, H. Wang, and Y.-X. Wang, "Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models," June 2024. arXiv:2310.04406 [cs].

[114] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, (Red Hook, NY, USA), pp. 9459–9474, Curran Associates Inc., Dec. 2020.

[115] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 6769–6781, Association for Computational Linguistics, Nov. 2020.

[116] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "REALM: retrieval-augmented language model pre-training," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119 of *ICML'20*, pp. 3929–3938, JMLR.org, July 2020.

[117] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, (New York, NY, USA), pp. 39–48, Association for Computing Machinery, July 2020.

[118] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, and M. Zaharia, "ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, eds.), (Seattle, United States), pp. 3715–3734, Association for Computational Linguistics, July 2022.

[119] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective Retrieval Augmented Generation," Oct. 2024. arXiv:2401.15884 [cs].

[120] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval," Jan. 2024. arXiv:2401.18059 [cs].

[121] R. Xu, H. Liu, S. Nag, Z. Dai, Y. Xie, X. Tang, C. Luo, Y. Li, J. C. Ho, C. Yang, and Q. He, "SimRAG: Self-Improving Retrieval-Augmented Generation for Adapting Large Language Models to Specialized Domains," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (L. Chiruzzo, A. Ritter, and L. Wang, eds.), (Albuquerque, New Mexico), pp. 11534–11550, Association for Computational Linguistics, Apr. 2025.

[122] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (Barcelona Spain), pp. 6491–6501, ACM, Aug. 2024.

[123] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," Oct. 2023.

[124] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark, "Self-Refine: Iterative Refinement with Self-Feedback," May 2023. arXiv:2303.17651 [cs].

[125] Q. Liu, H. Xu, X. Chen, W. Chen, Y. W. Teh, and N. Miao, "Enhancing Large Language Model Reasoning with Reward Models: An Analytical Survey," Oct. 2025. arXiv:2510.01925 [cs].

[126] M. Khalifa, R. Agarwal, L. Logeswaran, J. Kim, H. Peng, M. Lee, H. Lee, and L. Wang, "Process Reward Models That Think," Sept. 2025. arXiv:2504.16828 [cs].

[127] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, "Let's Verify Step by Step," May 2023. arXiv:2305.20050 [cs].

[128] M. He, Y. Shen, W. Zhang, Z. Tan, and W. Lu, "Advancing Process Verification for Large Language Models via Tree-Based Preference Learning," June 2024. arXiv:2407.00390 [cs].

[129] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang, "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," Oct. 2023. arXiv:2308.08155 [cs].

[130] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society," Nov. 2023.

[131] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, C. Zhang, J. Wang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, C. Wu, and J. Schmidhuber, "MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework," Nov. 2023.

[132] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, and Z. Liu, "ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate," Aug. 2023.

[133] K. Xiong, X. Ding, Y. Cao, T. Liu, and B. Qin, "Examining Inter-Consistency of Large Language Models Collaboration: An In-depth Analysis via Debate," in *Findings of the Association for Computational Linguistics: EMNLP 2023* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 7572–7590, Association for Computational Linguistics, Dec. 2023.

[134] H. Chen, W. Ji, L. Xu, and S. Zhao, "Multi-Agent Consensus Seeking via Large Language Models," Oct. 2023.

[135] P. Li, V. Menon, B. Gudiguntla, D. Ting, and L. Zhou, "Challenges Faced by Large Language Models in Solving Multi-Agent Flocking," Apr. 2024.

[136] D. M. Wegner, "Transactive Memory: A Contemporary Analysis of the Group Mind," in *Theories of Group Behavior* (B. Mullen and G. R. Goethals, eds.), pp. 185–208, New York, NY: Springer, 1987.

[137] K. Lewis, "Measuring transactive memory systems in the field: scale development and validation," *The Journal of Applied Psychology*, vol. 88, pp. 587–604, Aug. 2003.

[138] H. Shang, X. Liu, Z. Liang, J. Zhang, H. Hu, and S. Guo, "United Minds or Isolated Agents? Exploring Coordination of LLMs under Cognitive Load Theory," Sept. 2025. arXiv:2506.06843 [cs].

[139] S. Saadaoui and E. Alonso, "Coordinated LLM multi-agent systems for collaborative question-answer generation," *Knowledge-Based Systems*, vol. 330, p. 114627, Nov. 2025.

[140] N. Bienefeld, M. Kolbe, G. Camen, D. Huser, and P. K. Buehler, "Human-AI teaming: leveraging transactive memory and speaking up for enhanced team effectiveness," *Frontiers in Psychology*, vol. 14, Aug. 2023. Publisher: Frontiers.

[141] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Aug. 2019. arXiv:1908.10084 [cs].

[142] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," in *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, (,), pp. ,, ,, Aug. 2021.

[143] W. Sun, H. Cai, H. Chen, P. Ren, Z. Chen, M. de Rijke, and Z. Ren, "Answering Ambiguous Questions via Iterative Prompting," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 7669–7683, Association for Computational Linguistics, July 2023.

[144] S. Kim, J. Suk, S. Longpre, B. Y. Lin, J. Shin, S. Welleck, G. Neubig, M. Lee, K. Lee, and M. Seo, "Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, eds.), (Miami, Florida, USA), pp. 4334–4353, Association for Computational Linguistics, Nov. 2024.

[145] N. Lambert, V. Pyatkin, J. Morrison, L. Miranda, B. Y. Lin, K. Chandu, N. Dziri, S. Kumar, T. Zick, Y. Choi, N. A. Smith, and H. Hajishirzi, "RewardBench: Evaluating Reward Models for Language Modeling," in *Findings of the Association for Computational Linguistics: NAACL 2025* (L. Chiruzzo, A. Ritter, and L. Wang, eds.), (Albuquerque, New Mexico), pp. 1755–1797, Association for Computational Linguistics, Apr. 2025.

[146] O. Vasilyev, V. Dharnidharka, and J. Bohannon, "Fill in the BLANC: Human-free quality estimation of document summaries," in *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems* (S. Eger, Y. Gao, M. Peyrard, W. Zhao, and E. Hovy, eds.), (Online), pp. 11–20, Association for Computational Linguistics, Nov. 2020.

[147] K. Pillutla, S. Swayamdipta, R. Zellers, J. Thickstun, S. Welleck, Y. Choi, and Z. Harchaoui, "MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 4816–4828, Curran Associates, Inc., 2021.

[148] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment," May 2023.

[149] S. Kim, J. Shin, Y. Cho, J. Jang, S. Longpre, H. Lee, S. Yun, S. Shin, S. Kim, J. Thorne, and M. Seo, "Prometheus: Inducing Fine-grained Evaluation Capability in Language Models," Mar. 2024.

[150] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, S. Zhang, X. Deng, A. Zeng, Z. Du, C. Zhang, S. Shen, T. Zhang, Y. Su, H. Sun, M. Huang, Y. Dong, and J. Tang, "AgentBench: Evaluating LLMs as Agents," in *The Twelfth International Conference on Learning Representations*, pp. –, Oct. 2023.

[151] Meta AI, "Llama 3.1." Project blog, 2024. Accessed 2025-11-01.

[152] Qwen Team, "Qwen2.5: Technical report and release notes." Project blog, 2024. Accessed 2025-11-01.

[153] Google DeepMind, "Gemma 3 27b instruct." Model card, 2025. Accessed 2025-11-01.

[154] R. Artstein and M. Poesio, "Survey Article: Inter-Coder Agreement for Computational Linguistics," *Computational Linguistics*, vol. 34, no. 4, pp. 555–596, 2008.

[155] J. Cohen, "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, Apr. 1960. Publisher: SAGE Publications Inc.

[156] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, no. 5, pp. 378–382, 1971. Place: US Publisher: American Psychological Association.

[157] K. Krippendorff, "Computing krippendorff's alpha-reliability," in ,, 2011.

[158] A. Amiri-Margavi, I. Jebellat, E. Jebellat, and S. P. M. Davoudi, "Enhancing Answer Reliability Through Inter-Model Consensus of Large Language Models," Nov. 2024. arXiv:2411.16797 [cs] version: 1.

[159] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," ,, 1996.

[160] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, p. 205, Mar. 2017.

[161] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," Sept. 2011. arXiv:1109.2378 [stat].

[162] S. Sia, A. Dalmia, and S. J. Mielke, "Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 1728–1736, Association for Computational Linguistics, Nov. 2020.

[163] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," Feb. 2020.

[164] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 52, 1955.

[165] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal Sentence Encoder for English," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (E. Blanco and W. Lu, eds.), (Brussels, Belgium), pp. 169–174, Association for Computational Linguistics, Nov. 2018.

[166] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation," Feb. 2024.

[167] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards General Text Embeddings with Multi-stage Contrastive Learning," Aug. 2023. arXiv:2308.03281 [cs].

[168] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive Text Embedding Benchmark," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics* (A. Vlachos and I. Augenstein, eds.), (Dubrovnik, Croatia), pp. 2014–2037, Association for Computational Linguistics, May 2023.

[169] A. Rogers, O. Kovaleva, and A. Rumshisky, "A Primer in BERTology: What We Know About How BERT Works," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020. Place: Cambridge, MA Publisher: MIT Press.

[170] S. Agarwal, D. Wood, B. A. K. Murray, Y. Wei, A. A. Busaidi, S. Kafiabadi, E. Guilhem, J. Lynch, M. Townend, A. Mazumder, G. J. Barker, J. H. Cole, P. Sasieni, S. Ourselin, M. Modat, and T. C. Booth, "Impact of hospital-specific domain adaptation on BERT-based models to classify neuroradiology reports," *European Radiology*, vol. 35, pp. 5299–5313, Sept. 2025.

[171] S. Ghosh, S. Sengupta, S. Naskar, and S. K. Singh, "FinRAD: Financial Readability Assessment Dataset - 13,000+ Definitions of Financial Terms for Measuring Readability," in *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022* (M. El-Haj, P. Rayson, and N. Zmandar, eds.), (Marseille, France), pp. 1–9, European Language Resources Association, June 2022.

[172] "Finrad financial readability assessment dataset." https://huggingface.co/datasets/sohomghosh/FinRAD_Financial_Readability_Assessment_Dataset. Hugging Face Datasets, accessed 2025-11-09.

[173] A. V. Aho and M. J. Corasick, "Efficient string matching: an aid to bibliographic search," *Communications of the ACM*, vol. 18, pp. 333–340, June 1975.

[174] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," *Diploma thesis, Institut für Informatik, Technische Universität München*, Apr. 1991.

[175] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, pp. 157–166, Mar. 1994.

[176] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," Feb. 2013. arXiv:1211.5063 [cs].

[177] R. M. Neal, "Annealed Importance Sampling," Sept. 1998. arXiv:physics/9803008.

[178] A. Kong, J. S. Liu, and W. H. Wong, "Sequential Imputations and Bayesian Missing Data Problems," *Journal of the American Statistical Association*, vol. 89, pp. 278–288, Mar. 1994. Publisher: ASA Website _eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1994.10476469.

[179] J. S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, Sept. 1998. Publisher: ASA Website _eprint: https://doi.org/10.1080/01621459.1998.10473765.

[180] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, June 2016. ISSN: 1063-6919.

[181] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Jan. 2019. arXiv:1711.05101 [cs].

[182] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," May 2017. arXiv:1608.03983 [cs].

[183] "Fnspid-filtered-nasdaq-100." https://huggingface.co/datasets/benstaf/FNSPID-filtered-nasdaq-100. Hugging Face Datasets, accessed 2025-11-09.

[184] "Earnings calls qa." https://huggingface.co/datasets/lamini/earnings-calls-qa. Hugging Face Datasets, accessed 2025-11-09.

[185] "Finance alpaca / fiqa." https://huggingface.co/datasets/gbharti/finance-alpaca. Hugging Face Datasets, accessed 2025-11-09.

[186] "Finbro v0.1.0." https://huggingface.co/datasets/taddeusb90/finbro-v0.1.0. Hugging Face Datasets, accessed 2025-11-09.

[187] E. Levina and P. J. Bickel, "Maximum Likelihood estimation of intrinsic dimension," in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'04, (Cambridge, MA, USA), pp. 777–784, MIT Press, Dec. 2004.

[188] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, pp. 224–227, Apr. 1979.

[189] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics - Theory and Methods*, Jan. 1974. Publisher: Marcel Dekker, Inc.

[190] R. Li, Y. Wang, Z. Wen, M. Cui, and Q. Miao, "Different paths to the same destination: Diversifying LLMs generation for multi-hop open-domain question answering," *Knowledge-Based Systems*, vol. 309, p. 112789, Jan. 2025.

[191] S. Shen, Y. Li, N. Du, X. Wu, Y. Xie, S. Ge, T. Yang, K. Wang, X. Liang, and W. Fan, "On the Generation of Medical Question-Answer Pairs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 8822–8829, Apr. 2020.

[192] Z. Zeng, Q. Cheng, X. Hu, Y. Zhuang, X. Liu, K. He, and Z. Liu, "KoSEL: Knowledge subgraph enhanced large language model for medical question answering," *Knowledge-Based Systems*, vol. 309, p. 112837, Jan. 2025.

[193] G. Kumar, R. Banchs, and L. F. D'Haro, "RevUP: Automatic Gap-Fill Question Generation from Educational Texts," in *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications* (J. Tetreault, J. Burstein, and C. Leacock,

eds.), (Denver, Colorado), pp. 154–161, Association for Computational Linguistics, June 2015.

[194] M. Uto, Y. Tomikawa, and A. Suzuki, "Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory," in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)* (E. Kochmar, J. Burstein, A. Horbach, R. Laarmann-Quante, N. Madnani, A. Tack, V. Yaneva, Z. Yuan, and T. Zesch, eds.), (Toronto, Canada), pp. 119–129, Association for Computational Linguistics, July 2023.

[195] Y. Meng, L. Pan, Y. Cao, and M.-Y. Kan, "FollowupQG: Towards information-seeking follow-up question generation," 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 (Volume 1: Long Papers)* (J. C. Park, Y. Arase, B. Hu, W. Lu, D. Wijaya, A. Purwarianti, and A. A. Krisnadhi, eds.), (Nusa Dua, Bali), pp. 252–271, Association for Computational Linguistics, Nov. 2023.

[196] X. Yuan, T. Wang, Y.-H. Wang, E. Fine, R. Abdelghani, H. Sauzéon, and P.-Y. Oudeyer, "Selecting Better Samples from Pre-trained LLMs: A Case Study on Question Generation," in *Findings of the Association for Computational Linguistics: ACL 2023* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 12952–12965, Association for Computational Linguistics, July 2023.

[197] Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang, and Q. Li, "Recommender Systems in the Era of Large Language Models (LLMs)," *IEEE Trans. on Knowl. and Data Eng.*, vol. 36, pp. 6889–6907, Nov. 2024.

[198] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang, "Large Language Model based Multi-Agents: A Survey of Progress and Challenges," Apr. 2024.

[199] M. L. Blanton and D. A. Stylianou, "Understanding the role of transactive reasoning in classroom discourse as students learn to construct proofs," *The Journal of Mathematical Behavior*, vol. 34, pp. 76–98, June 2014.

[200] A. W. Woolley and P. Gupta, "Understanding collective intelligence: Investigating the role of collective memory, attention, and reasoning processes," *Perspectives on Psychological Science*, vol. 19, pp. 344 – 354, 2023.

[201] E. Kamalloo, N. Dziri, C. Clarke, and D. Rafiei, "Evaluating Open-Domain Question Answering in the Era of Large Language Models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 5591–5606, Association for Computational Linguistics, July 2023.

[202] D. J. Hall and R. A. Davis, "Engaging multiple perspectives: A value-based decision-making model," *Decision Support Systems*, vol. 43, pp. 1588–1604, Aug. 2007.

[203] Y. Shao, Y. Jiang, T. A. Kanell, P. Xu, O. Khattab, and M. S. Lam, "Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models," Apr. 2024.

[204] M. Ihori, H. Sato, T. Tanaka, and R. Masumura, "Multi-Perspective Document Revision," in *Proceedings of the 29th International Conference on Computational Linguistics* (N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, and S.-H. Na, eds.), (Gyeongju, Republic of Korea), pp. 6128–6138, International Committee on Computational Linguistics, Oct. 2022.

[205] I. Momennejad, "Collective minds: Social network topology shapes collective cognition," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 377, p. 20200315, Dec. 2021.

[206] H. V. D. Parunak, T. C. Belding, R. Hilscher, and S. A. Brueckner, "Cognitive collapse: Recognizing and addressing the hidden threat in collaborative technologies," in *Defense Technical Information Center*, (,), pp. ,, ,, 2008.

[207] H. V. Parunak, T. C. Belding, R. Hilscher, and S. Brueckner, "Understanding Collective Cognitive Convergence," in *Multi-Agent-Based Simulation IX* (N. David and J. S. Sichman, eds.), (Berlin, Heidelberg), pp. 46–59, Springer, 2009.

[208] S. D. Teasley, F. Fischer, A. Weinberger, K. Stegmann, P. Dillenbourg, M. Kapur, and M. Chi, "Cognitive convergence in collaborative learning," in *Proceedings of the 8th International Conference on International Conference for the Learning Sciences - Volume 3*, ICLS'08, (Utrecht, The Netherlands), pp. 360–367, International Society of the Learning Sciences, June 2008.

[209] A. W. Woolley, I. Aggarwal, and T. W. Malone, "Collective Intelligence and Group Performance," *Current Directions in Psychological Science*, vol. 24, no. 6, pp. 420–424, 2015.

[210] A. W. Woolley and P. Gupta, "Understanding Collective Intelligence: Investigating the Role of Collective Memory, Attention, and Reasoning Processes," *Perspectives on Psychological Science*, vol. 19, pp. 344–354, Mar. 2024.

[211] D. Friedman and A. B. Dieng, "The Vendi Score: A Diversity Evaluation Metric for Machine Learning," July 2023.

[212] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, M. Rosenberg, X. Song, A. Stoica, S. Tiwary, and T. Wang, "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset," Oct. 2018.

[213] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural Questions: A Benchmark for Question Answering Research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 452–466, 2019.

[214] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur, "WWW'18 Open Challenge: Financial Opinion Mining and Question Answering," in *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, (Lyon, France), pp. 1941–1942, ACM Press, 2018.

[215] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying deep learning to answer selection: A study and an open task," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (Scottsdale, AZ, USA), pp. 813–820, IEEE, Dec. 2015.

[216] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, "What disease does this patient have? a large-scale open domain question answering dataset from medical exams," *Applied Sciences*, vol. 11, no. 14, p. 6421, 2021.

[217] A. Pal, L. K. Umapathi, and M. Sankarasubbu, "MedMCQA: A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering," in *Proceedings of the Conference on Health, Inference, and Learning*, pp. 248–260, PMLR, Apr. 2022. ISSN: 2640-3498.

[218] L. AI, "Langgraph (repository)," 2024.

[219] Anonymous, "Cir3: Collective intentional reading through reflection and refinement (repository)," 2025.

[220] Groq, "Groq: Ai inference," 2024.

[221] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, "Efficient Memory Management for Large Language Model Serving with PagedAttention," in *Proceedings of the 29th Symposium on Operating Systems Principles*, (Koblenz Germany), pp. 611–626, ACM, Oct. 2023.

[222] vLLM Project, "vllm (repository)," 2023.

[223] 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.

[224] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, eds.), (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June 2005.

[225] L. da Fontoura Costa, "Further generalizations of the jaccard index," *ArXiv*, vol. abs/2110.09619, pp. –, 2021.

[226] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235 of *ICML'24*, (Vienna, Austria), pp. 11733–11763, JMLR.org, July 2024.

[227] X. Zeng and A. Zubiaga, "Aggregating pairwise semantic differences for few-shot claim verification," *PeerJ Computer Science*, vol. 8, p. e1137, Oct. 2022.

[228] C. M. Giannantonio, "Book Review: Krippendorff, K. (2004). Content Analysis: An Introduction to Its Methodology (2nd ed.). Thousand Oaks, CA: Sage," *Organizational Research Methods*, vol. 13, pp. 392–394, Apr. 2010. Publisher: SAGE Publications Inc.

[229] H. Blidh and contributors, "simpledorff: Krippendorff's alpha for python," 2019.

[230] G. Marzi, M. Balzano, and D. Marchiori, "K-Alpha Calculator–Krippendorff's Alpha Calculator: A user-friendly tool for computing Krippendorff's Alpha inter-rater reliability coefficient," *MethodsX*, vol. 12, p. 102545, June 2024.

[231] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or fiction: Verifying scientific claims," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), (Online), pp. 7534–7550, Association for Computational Linguistics, Nov. 2020.

[232] Groq, "On-demand pricing for tokens-as-a-service," 2025. Accessed January 2025.