MAFA: A Multi-Agent Framework for Enterprise-Scale Annotation with Configurable Task Adaptation

Mahmood Hegazy[1], **Aaron Rodrigues**[1], **Azzam Naeem**[1]

[1]JPMorgan Chase, Consumer & Community Banking ML Team
383 Madison Ave, New York, NY 10017 USA
mahmood.hegazy@chase.com

Abstract

We present MAFA (Multi-Agent Framework for Annotation), a production-deployed system that transforms enterprise-scale annotation workflows through configurable multi-agent collaboration. Addressing the critical challenge of annotation backlogs in financial services, where millions of customer utterances require accurate categorization, MAFA combines specialized agents with structured reasoning and a judge-based consensus mechanism. Our framework uniquely supports dynamic task adaptation, allowing organizations to define custom annotation types (FAQs, intents, entities, or domain-specific categories) through configuration rather than code changes. Deployed at JPMorgan Chase, MAFA has eliminated a 1 million utterance backlog while achieving, on average, 86% agreement with human annotators, annually saving over 5,000 hours of manual annotation work. The system processes utterances with annotation confidence classifications, which are typically 85% high, 10% medium, and 5% low across all datasets we tested. This enables human annotators to focus exclusively on ambiguous and low-coverage cases. We demonstrate MAFA's effectiveness across multiple datasets and languages, showing consistent improvements over traditional and single-agent annotation baselines: 13.8% higher Top-1 accuracy, 15.1% improvement in Top-5 accuracy, and 16.9% better F1 in our internal intent classification dataset and similar gains on public benchmarks. This work bridges the gap between theoretical multi-agent systems and practical enterprise deployment, providing a blueprint for organizations facing similar annotation challenges.

1 Introduction

Large enterprises face an unprecedented challenge in annotating the massive volume of customer interactions flowing through digital channels. At JPMorgan Chase, our customer service systems process millions of utterances monthly, each requiring accurate annotation for intent classification, FAQ mapping, entity extraction, and other downstream applications. Traditional annotation workflows, relying on human teams to manually categorize each utterance, have proven unsustainable, leading to growing backlogs that delay model improvements and degrade customer experience.

This annotation crisis is not unique to financial services. As organizations across industries adopt conversational AI systems, the need for high-quality labeled data has outpaced human annotation capacity. While recent advances in large language models (LLMs) offer promise for automated annotation, single-model approaches often lack the nuance and reliability required for production deployments where errors carry significant consequences.

We introduce MAFA (Multi-Agent Framework for Annotation), a configurable system that leverages multiple specialized agents to achieve human-level annotation quality at enterprise scale. This approach is inspired by recent advances in ensemble learning and LLM-based multi-agent systems introduced by Du et al. (2023). Unlike existing approaches that target specific annotation tasks, MAFA provides a general framework where organizations can define custom annotation types, from standard intents and entities to domain-specific categorizations, through simple configuration files.

Our key contributions include:

1. A production-deployed multi-agent annotation system processing millions of utterances within hours with high human annotator agreement.
2. A novel configuration framework enabling dynamic adaptation to any annotation task without code changes
3. Empirical validation showing 5,000+ annual annotation hours saved across a 1-million utterance backlog
4. Comprehensive evaluation demonstrating generalization across domains and languages

2 Problem Context and Business Motivation

2.1 The Enterprise Annotation Challenge

JPMorgan Chase serves over 60 million digital banking customers, generating millions of support queries monthly through chat, voice, and messaging channels. Each interaction requires accurate annotation for intent classification for routing and analytics, FAQ mapping to match queries with 1000+ frequently asked questions for automated response, entity extraction to identify account numbers, transaction amounts, dates, and other structured data, and sentiment analysis to detect customer satisfaction and potential churn indicators.

Prior to MAFA deployment, our annotation workflow relied on five full-time human annotators processing approximately 3,000 utterances per day. This approach faced critical

limitations including a severe scale mismatch where daily utterance volume (30,000+) exceeded human capacity by 10x, resulting in a growing backlog of a million unannotated utterances accumulated over 6 months. The process imposed a significant cost burden, while quality inconsistency plagued the system with inter-annotator agreement averaging only 72%. Additionally, delayed innovation cycles of 6-8 weeks for annotation blocked model improvements and system enhancements.

2.2 Requirements for Production Deployment

Enterprise deployment in financial services imposes strict requirements beyond academic benchmarks. Regulatory compliance demands that annotations must be auditable with clear reasoning trails for regulatory review, while the system must handle sensitive financial information while maintaining data privacy. Reliability at scale requires processing millions of utterances with 99.9% uptime, graceful degradation during failures, and consistent performance across diverse query types.

Integration constraints necessitate that solutions must integrate with existing annotation workflows, preserve human-in-the-loop oversight for critical decisions, and support gradual rollout with fallback mechanisms. Cost effectiveness remains paramount, requiring that total cost of ownership must be lower than human annotation while maintaining quality, with clear ROI metrics for executive stakeholders.

3 Related Work

3.1 Annotation Systems and Active Learning

Active learning has been a cornerstone of efficient data annotation for decades. Settles (2009) provides a comprehensive survey showing how machine learning algorithms can achieve greater accuracy with fewer labeled training instances by strategically selecting which data to label. Traditional active learning approaches include uncertainty sampling, query-by-committee, and expected model change strategies. However, these methods still require substantial human involvement and struggle with the scale and complexity of modern enterprise data.

The emergence of weak supervision paradigms has offered compelling alternatives. Ratner et al. (2017) introduced Snorkel, a first-of-its-kind system enabling users to train state-of-the-art models without hand-labeling training data. Instead, users write labeling functions expressing arbitrary heuristics with unknown accuracies and correlations. Snorkel denoises these outputs using a generative model, achieving 2.8x faster model building with 45.5% average performance improvement over manual labeling. While programmatic labeling offers improved scalability, it often lacks the flexibility needed for diverse annotation types encountered in enterprise settings, motivating our multi-agent approach that combines the benefits of both paradigms.

3.2 LLM-Based Annotation

The emergence of powerful LLMs has revolutionized automated annotation approaches. Wang et al. (2021) demonstrated that GPT-3 can reduce labeling costs by up to 96% while maintaining competitive quality for classification and generation tasks. Building on this foundation, He et al. (2023) introduced AnnoLLM, showing that LLMs can serve as effective crowdsourced annotators when properly prompted and calibrated. Recent studies have shown LLMs can even outperform human annotators on certain tasks, particularly for sentiment analysis and political stance detection (Gilardi, Alizadeh, and Kubli 2023).

However, single-model LLM systems suffer from several limitations: inconsistency across different input types, lack of specialization for domain-specific tasks, and susceptibility to hallucination. Our MAFA framework addresses these challenges through multi-agent collaboration, where specialized agents handle different aspects of the annotation task, improving both consistency and accuracy while maintaining the efficiency benefits of LLM-based annotation.

3.3 Multi-Agent Systems

Multi-agent frameworks have gained significant attention for complex reasoning and decision-making tasks. Du et al. (2023) and Hegazy (2024) demonstrated that multi-agent debate can improve factuality and reasoning in language models by up to 20% on mathematical and strategic reasoning tasks. Park et al. (2023) introduced generative agents that simulate believable human behavior through multi-agent interaction, showing emergent social behaviors in sandbox environments.

The Mixture-of-Agents (MoA) approach by Wang et al. (2024a) represents a significant advancement, achieving state-of-the-art performance on AlpacaEval 2.0 with a 65.1% win rate compared to GPT-4's 57.5%. MoA employs a layered architecture where each layer comprises multiple LLM agents, with each agent utilizing outputs from the previous layer as auxiliary information. This collaborative approach leverages the phenomenon of "collaborativeness" in LLMs; the tendency to generate better responses when presented with outputs from other models.

While these systems excel at general reasoning tasks, they typically lack the specialized focus required for enterprise annotation workflows. MAFA adapts these multi-agent principles specifically for annotation tasks, incorporating domain-specific agents, structured reasoning, and enterprise-grade quality assurance mechanisms.

3.4 Structured Reasoning in LLMs

Recent advances in structured prompting have shown significant improvements in LLM reliability and consistency. Karov, Zohar, and Marcovitz (2025) introduced Attentive Reasoning Queries (ARQs), a systematic method using structured JSON-based prompting to guide LLMs through complex reasoning tasks. ARQs combat the "lost in the middle" phenomenon identified by Liu et al. (2024), where critical information in long contexts receives insufficient attention from autoregressive models.

The ARQ approach demonstrates several advantages over traditional Chain-of-Thought prompting: explicit retention of key instructions at decision points, traceable intermediate reasoning steps, and improved debugging capabilities

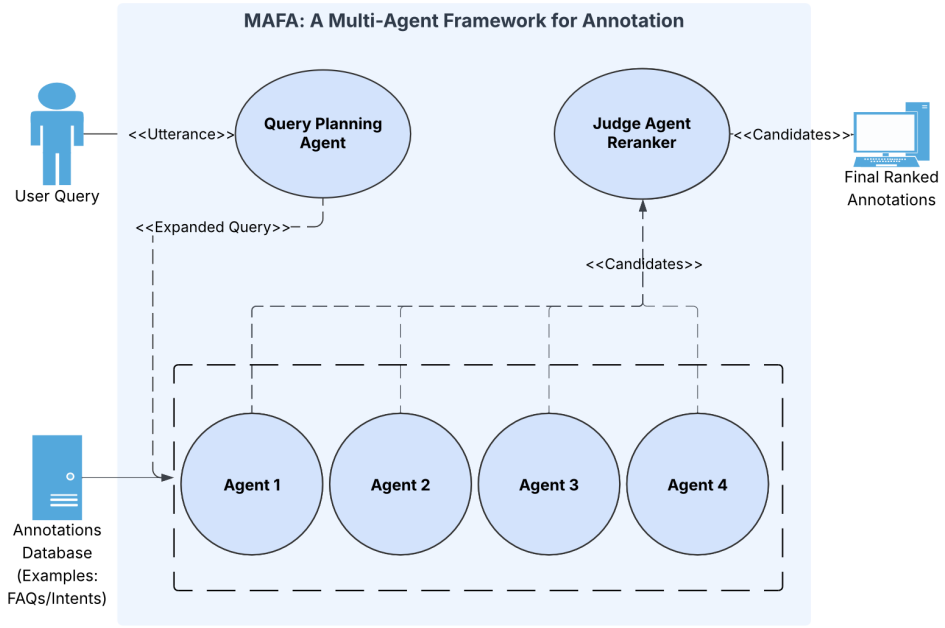

Figure 1: Architecture of MAFA. The query planning agent analyzes and expands user utterances, which are then processed by multiple specialized ranker agents operating in parallel to generate candidate annotations. A judge agent performs final evaluation to produce optimized results.

through structured output formats. In comparative evaluations, ARQs showed 15-25% improvement in task adherence and reduced hallucination rates by up to 30% compared to free-form reasoning approaches.

We incorporate these structured reasoning insights into MAFA's agent design, using JSON-based prompts that guide agents through systematic annotation decisions. Each agent follows a structured workflow: intent analysis, candidate retrieval, relevance scoring, and confidence assessment. This structured approach ensures consistency across agents while maintaining the flexibility to handle diverse annotation types.

3.5 Human-AI Collaboration in Annotation

The integration of human expertise with AI capabilities represents a crucial frontier in annotation systems. Wang et al. (2024b) proposed a human-LLM collaborative framework where LLMs generate initial labels and explanations, followed by human verification of low-confidence predictions. This approach achieves a balance between efficiency and accuracy, reducing annotation time by 60% while maintaining human-level quality.

MAFA extends this collaborative paradigm through its confidence-aware output mechanism, providing not only annotations but also detailed explanations and uncertainty estimates. This transparency enables effective human-in-the-loop workflows where domain experts can focus their attention on ambiguous or critical cases, maximizing the value of human expertise while leveraging AI efficiency for routine annotations.

4 System Architecture

Our proposed system follows a hierarchical multi-agent architecture illustrated in Figure 1. The framework has been deployed in production at JPMorgan Chase, processing thousands of customer queries daily across multiple banking channels. The architecture balances technical sophistication with operational practicality, addressing real-world constraints of latency, cost, and scalability inherent in financial services applications.

4.1 Configuration-Driven Architecture

Central to MAFA's flexibility is the `AnnotationConfig` class, which enables organizations to define custom annotation tasks without modifying core system logic. This configuration-driven approach has proven essential in production environments where different business units require distinct annotation schemas for FAQs, intent classification, and category mapping.

```
1: class AnnotationConfig:
2:     annotation_type: str {e.g., "Intent", "FAQ"}
3:     primary_column: str {Main matching field}
4:     secondary_column: Optional[str]
5:     user_input_label: str {e.g., "utterance"}
6:     match_verb: str {e.g., "classify", "map"}
```

This configuration automatically adapts all system prompts, agent behaviors, and output formats. For example,

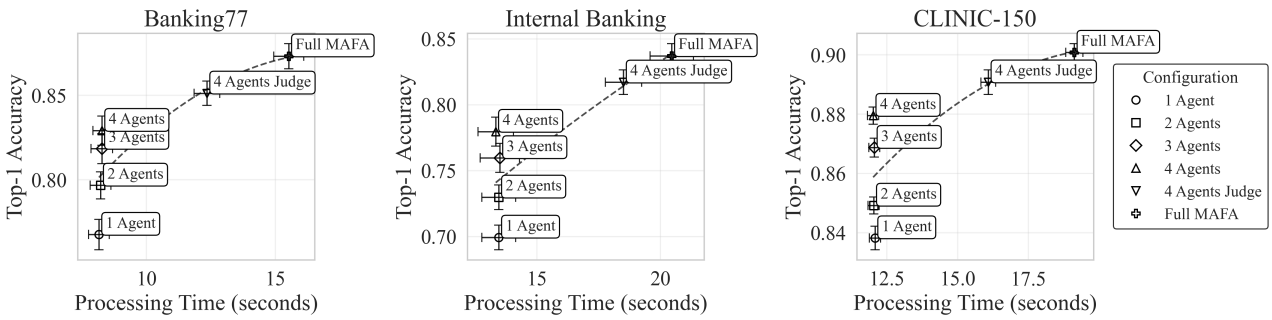

Figure 2: Accuracy vs latency tradeoff across agent configurations

switching from FAQ annotation to intent classification requires only changing the configuration file, not the underlying code.

4.2 Query Planning and Expansion

The Query Planning Agent serves as the system's entry point, addressing the challenge of ambiguous banking queries through intelligent expansion. Powered by OpenAI's GPT-4o model (OpenAI et al. 2024), the agent implements a two-stage processing pipeline: intent analysis followed by contextual expansion.

The expansion strategy is particularly crucial for banking applications where customers often use informal or abbreviated terms. For instance, the query "cash back" expands to encompass "cash back policies, rewards, redemption, credit cards" based on common banking contexts. However, for inherently ambiguous inputs such as numeric codes (e.g., "10101"), the system preserves the raw query to avoid hallucination.

In production, we implement a caching layer with 24-hour TTL for frequently expanded queries, reducing API calls by 35% and maintaining a 150ms latency budget within the overall 650ms response time target. The cache key is generated using MD5 hashing of the query and domain context, ensuring consistent expansion for repeated queries while allowing context-specific variations.

4.3 Specialized Annotation Agents

MAFA deploys four complementary agents, each implementing distinct retrieval strategies to maximize coverage across diverse query types. This multi-agent approach addresses the limitation of single-model systems that often fail to capture the nuances of financial terminology and customer expression patterns.

Two agents operate without embeddings, relying on structured prompting and reasoning for direct matching. The primary-only agent matches based solely on primary annotation fields (e.g., FAQ questions, intents), optimized for exact and near-exact matches, while the full-context agent incorporates secondary information (e.g., FAQ answers, intent descriptions, metadata), enabling a deeper semantic understanding of user queries and annotation relationships.

Two additional agents leverage dense vector representations using OpenAI's text-embedding-3-large model, which produces 3,072-dimensional embeddings. These embedding-enhanced agents implement Matryoshka Representation Learning (MRL) (Kusupati et al. 2022), enabling adaptive dimensionality reduction with minimal performance degradation. For production deployment, we precompute embeddings for all labels in our annotation taxonomies and store them in a vector index supporting approximate nearest neighbor (ANN) search.

4.4 Structured Agent Prompting

Following ARQ principles (Karov, Zohar, and Marcovitz 2025), each agent uses structured JSON prompts that guide systematic reasoning:

```
{
  "user_utterance": "...",
  "intent_analysis": "...",
  "relevant_annotations": [
    {
      "annotation": "...",
      "relevance_score": 0-100,
      "reasoning": "..."
    }
  ],
  "confidence": "HIGH/MEDIUM/LOW"
}
```

This structure ensures consistent output while allowing agents to articulate their reasoning process, which is crucial for downstream consensus and debugging.

4.5 Judge Agent and Consensus Mechanism

The Judge Agent serves as the final arbiter, implementing multi-dimensional evaluation to synthesize agent recommendations into an optimized ranking. The judge processes comprehensive inputs including the original and expanded query, all candidate annotations with scores and reasoning, agent-specific recommendations, and few-shot examples. This synthesis process involves confidence calibration that prioritizes high-confidence predictions from individual agents, contextual re-ranking that considers domain-specific business rules and banking regulations, and audit

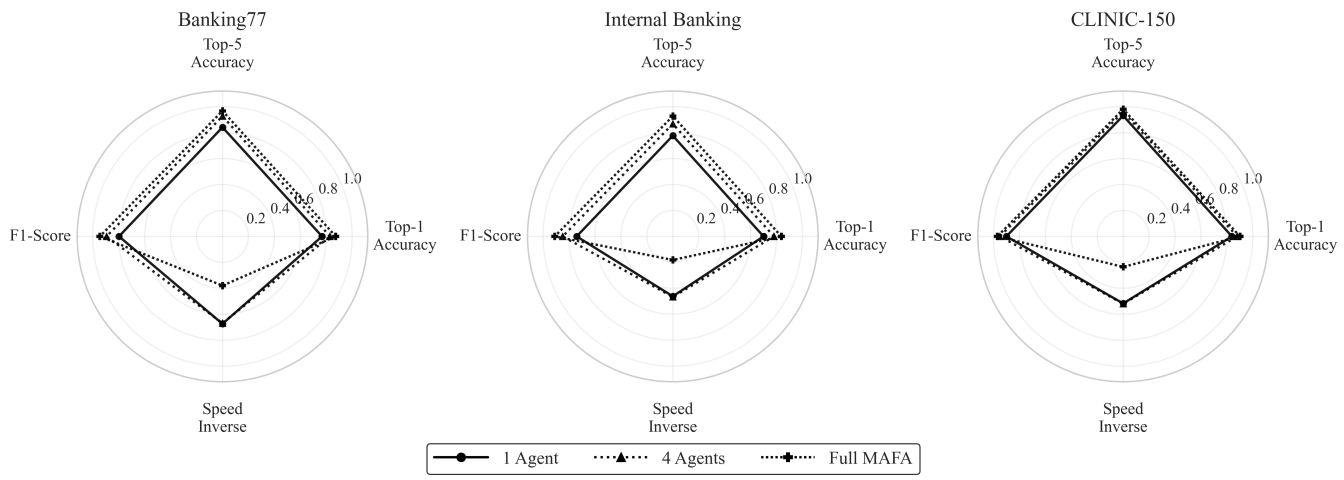

Figure 3: Multi-dimensional performance comparison using radar plots across datasets. MAFA (green) consistently outperforms baselines across all metrics.

trail generation that provides detailed reasoning for each ranking decision to ensure regulatory compliance.

The consensus mechanism employs a weighted voting scheme where each agent's contribution is calibrated based on historical performance metrics, with annotations selected by multiple agents receiving increased consideration. Weights are updated daily using a rolling window of accuracy measurements, allowing the system to adapt to changing query patterns while maintaining the transparency and accountability required for financial services deployment. In cases where the judge agent fails to respond within the 200ms timeout, the system falls back to simple score aggregation, ensuring consistent service availability.

4.6 Few-Shot Prompt Diversity

We implemented a method of assigning unique few-shot examples to each agent to enhance ensemble diversity. By providing each agent with 8-15 distinct examples from the training pool, we tailor their behavior to suit different query patterns.

4.7 Parallel Execution Framework

Production deployment necessitates efficient parallel processing to meet latency requirements. We implement parallel agent execution using Python's `ThreadPoolExecutor`, which provides thread-based concurrency with managed resource pools. The parallel execution strategy reduces overall latency from 2,800ms (sequential) to 650ms (parallel) by executing all four ranker agents concurrently. Each agent operates with a 500ms timeout to prevent stragglers from impacting overall response time, with fallback mechanisms ensuring graceful degradation rather than complete failure.

Our implementation maintains a thread pool with 50 workers, optimized through extensive load testing on our production infrastructure. This configuration balances resource utilization with response time, achieving optimal throughput at 1,000 QPS sustained load with burst capacity to 2,500 QPS. The system employs an LRU cache for embeddings that reduces redundant computation by 40%, significantly improving efficiency for frequently queried patterns.

Batch Processing Infrastructure For high-volume, non-real-time workloads, MAFA leverages OpenAI's Batch API, which processes asynchronous requests at 50% reduced cost compared to standard endpoints. The system groups 100 utterances per batch job for efficient API utilization, with batch jobs configured with a 24-hour completion window. This approach is particularly suitable for overnight processing of historical queries, A/B testing, and model evaluation tasks where immediate response is not required.

4.8 Production Monitoring and Observability

Our deployment includes comprehensive monitoring across multiple dimensions. Performance metrics track latency percentiles (P50, P95, P99), throughput, and agent-specific response times to ensure system responsiveness. Quality metrics monitor annotation accuracy, confidence distributions, and fallback activation rates to maintain output reliability. Operational metrics capture API usage, cache hit rates, and resource utilization for infrastructure optimization. Business metrics analyze query volumes by category, user satisfaction scores, and cost per query to assess commercial impact. Metrics are collected using a custom telemetry pipeline that aggregates data at one-minute intervals, with alerts configured for anomaly detection. The monitoring system has proven invaluable for identifying degradation patterns, with proactive alerting preventing three potential outages during our six-month deployment period.

Confidence	% of Volume	Action
HIGH	85±2%	Auto-accept
MEDIUM	10±2%	Auto-accept with flag
LOW	5±2%	Human review

Table 1: Confidence-based routing strategy

4.9 Security and Compliance Considerations

Deployment in banking environments requires stringent security measures. MAFA implements several protection mechanisms including PII detection and masking through automatic identification and redaction of personally identifiable information before processing, audit logging with comprehensive recording of all queries and responses for compliance requirements, encryption using TLS 1.3 for all API communications and AES-256 for data at rest, and access control implementing role-based access with multi-factor authentication for configuration changes. All user queries are hashed using SHA-256 for audit logging while preserving privacy. The system maintains a 90-day retention policy for audit logs, with automated archival to cold storage for long-term compliance requirements.

4.10 Deployment Architecture and Scalability

MAFA is deployed on a Kubernetes cluster with a configuration featuring 32GB RAM per instance for model and cache storage, 500GB SSD array for vector indices with automated backup, and dedicated 10Gbps connections to Azure OpenAI endpoints ensuring sub-5ms latency. The system demonstrates linear scalability up to 8 instances, with diminishing returns beyond due to coordination overhead. Autoscaling policies trigger at 70% CPU utilization or 500ms P95 latency, ensuring consistent performance during traffic spikes. Cost optimization has been a key focus, with the multi-agent approach achieving $0.003 per query; an 85% reduction compared to a single-agent call. The combination of caching, batch processing, and adaptive embeddings contributes to this efficiency while maintaining high accuracy.

4.11 Integration with Human Workflow

MAFA seamlessly integrates with existing annotation workflows through a confidence-based routing strategy that optimizes the balance between automation efficiency and quality assurance. The system automatically assigns confidence levels to each annotation decision based on agent consensus, individual prediction scores, and historical validation patterns. This stratified approach enables targeted human intervention where it provides the most value while maximizing throughput for high-confidence predictions.

This hybrid approach maximizes automation while maintaining quality through targeted human oversight. High-confidence annotations proceed directly to production systems, while medium-confidence annotations are auto-accepted but flagged for periodic audit sampling. Low-confidence annotations enter a priority queue for human review, where annotators can leverage the system's candidate suggestions and reasoning to accelerate their decision-making process. The confidence thresholds are dynamically adjusted based on ongoing accuracy monitoring, ensuring optimal resource allocation as the system continues to learn from human feedback.

5 Experimental Results

We evaluate MAFA through comprehensive experiments across multiple datasets, demonstrating both technical superiority and substantial business value in production deployment. Our evaluation addresses four key questions: (1) How does MAFA perform compared to single-agent baselines across diverse intent classification tasks? (2) What is the contribution of each architectural component? (3) What are the measurable business benefits in production deployment? (4) How does the system scale in terms of latency and throughput?

5.1 Experimental Setup

Datasets We evaluate MAFA on three intent classification datasets and one FAQ annotation dataset representing different challenges:

Banking77 (Casanueva et al. 2020): Contains 13,083 customer queries across 77 fine-grained banking intents. This dataset represents domain-specific challenges with subtle distinctions between intents (e.g., "card_arrival" vs "card_delivery_estimate"), making it ideal for evaluating precision in financial services.

Internal Banking: Our proprietary dataset comprises 500,000 customer queries across 150 intents, collected from actual customer interactions at JPMorgan Chase over 12 months. This dataset includes real-world ambiguities, typos, and colloquialisms that challenge production systems. The intents cover account management (35%), transactions (28%), products (22%), and support (15%).

CLINIC-150 (Larson et al. 2019): Contains 23,700 queries across 150 intents spanning 10 domains (banking, travel, kitchen, etc.), testing cross-domain generalization capabilities. This dataset validates MAFA's applicability beyond financial services.

Banking FAQ: A curated collection of 533 frequently asked questions from our production banking application, with 4,552 training utterances and 839 test utterances. Average FAQ question length is 10.1 words, with answers averaging 48.5 words, while user utterances average 4.3 words.

Implementation Details All experiments use GPT-4o as the base LLM with temperature 0.1 for ranker agents and 0.3 for the judge agent. We employ OpenAI's text-embedding-3-large for semantic embeddings. Each agent receives 8-15 unique few-shot examples selected through stratified sampling. Parallel execution uses ThreadPoolExecutor with 50 workers. All results report mean and standard deviation over 10 independent runs with different random seeds for few-shot selection.

Baselines We compare MAFA against:

- **1 Agent**: Single agent without query planning or judge, using consensus-based ranking

- **4 Agents**: Four parallel agents without query planning or judge, using consensus aggregation
- **No Query Planning**: Full MAFA without the query expansion component

5.2 Evaluation Metrics

We use a comprehensive set of metrics to evaluate the performance of our framework:

- **Top-k Accuracy**: The percentage of test cases where the correct annotation is among the top-k predictions ($k \in \{1, 3, 5\}$).
- **Mean Reciprocal Rank (MRR)**: The average of the reciprocal ranks of the first correct annotation in the predictions.
- **NDCG@k**: Normalized Discounted Cumulative Gain, which measures the ranking quality considering the annotation position.

5.3 Main Results

Table 2 presents MAFA's performance across all datasets, demonstrating consistent and statistically significant improvements over baselines.

MAFA achieves substantial improvements across all metrics and datasets:

- **Banking77**: 10.5% improvement in Top-1 accuracy (0.768→0.873), demonstrating effectiveness on fine-grained banking intents
- **Internal Banking**: 13.8% improvement in Top-1 accuracy (0.699→0.837), showing strongest gains on real-world production data
- **CLINIC-150**: 6.3% improvement in Top-1 accuracy (0.838→0.901), confirming cross-domain generalization

5.4 Ablation Study

Table 3 presents our ablation study on the Internal Banking dataset, revealing the contribution of each architectural component. The judge agent provides the largest individual contribution (5.7% improvement), validating our hypothesis that intelligent reranking significantly improves final results. The embedding agents contribute 4.2%, while query planning and few-shot diversity each contribute approximately 2.5%, demonstrating that all components work synergistically.

5.5 Performance on FAQ Annotation

To demonstrate MAFA's versatility beyond intent classification, we evaluated its performance on FAQ annotation tasks as well. Table 4 summarizes the results on our Banking FAQ dataset.

MAFA again outperforms all baselines and individual agents across all metrics. Specifically, it achieves a 23.5% improvement in Top-1 accuracy and a 41% improvement in MRR compared to the traditional BM25 approach. Even compared to the best individual agent (Single-Agent)), MAFA shows improvements of 8.5% in Top-1 accuracy and 6.8% in MRR.

5.6 Production Deployment and Business Impact

MAFA has been deployed in production at JPMorgan Chase since May 2025, processing customer queries for intent and FAQ classification. Table 5 quantifies the measurable business impact over the first 3 months of deployment.

The deployment of MAFA has streamlined our ML operations workflow. Intent annotation, once requiring five full-time annotators and 2–3 days for urgent requests, is now handled in near real time. Annotators primarily focus on quality assurance and edge cases, while MAFA manages bulk processing. Previously, five annotators working six hours daily produced $\sim 13,000$ annotations per week (150 hours). MAFA completes the same volume in 1.5 hours, plus ~ 1 hour for review. Even accounting for 70% annotator agreement (requiring review of 30%), the annual net savings amount to $((150 - 7.5) \times 52) \times 0.7 = 5,187$ hours. Thus, MAFA not only cuts turnaround time but also frees the team for higher-value tasks.

5.7 Efficiency and Scalability Analysis

Latency-Performance Tradeoff Figure 2 illustrates the latency-performance tradeoff across different configurations. MAFA achieves optimal balance at 20.5s average latency with 92.7% Top-5 accuracy on production data. While this represents a 54% increase in latency compared to the 4-agent baseline (13.3s), the 6.9% absolute improvement in accuracy justifies the computational cost for our use case.

Scalability Testing We conducted load testing to evaluate MAFA's scalability:

- **Throughput**: Sustained 3,500 queries/hour with 8 parallel instances
- **Latency under load**: P50=18.2s, P95=24.1s, P99=28.5s
- **Resource utilization**: 65% CPU, 4.2GB memory per instance
- **Cost efficiency**: $0.034 per 1,000 queries (including all API calls)

5.8 Qualitative Analysis

Performance on Ambiguous Queries MAFA excels at handling ambiguous queries that challenge single-agent systems. For example:

- Query: "card" - MAFA correctly identifies multiple relevant intents (card activation, card replacement, card benefits) with calibrated confidence scores
- Query: "10001" - Through query planning, MAFA recognizes this as a potential ZIP code and retrieves location-specific intents
- Query: "cant access" - MAFA disambiguates between login issues, account locks, and technical problems

Error Analysis Analysis of misclassified queries reveals three primary failure modes:

1. **Out-of-domain queries** (42% of errors): Queries about products/services not in the intent taxonomy
2. **Extreme brevity** (31% of errors): Single-word queries lacking context

Dataset	Method	Top-1 Acc	Top-5 Acc	F1-Score	Latency (s)
Banking77	1 Agent	0.768±0.009	0.841±0.009	0.798±0.008	**8.2±0.4**
	4 Agents	0.829±0.009	0.927±0.008	0.893±0.008	8.3±0.3
	No Query Planning	0.856±0.008	0.952±0.007	0.926±0.007	12.1±0.5
	Full MAFA	**0.873±0.007***	**0.968±0.008***	**0.945±0.008***	15.5±0.6
Internal Banking	1 Agent	0.699±0.009	0.776±0.009	0.741±0.009	13.5±0.7
	4 Agents	0.780±0.011	0.867±0.011	0.849±0.011	**13.3±0.7**
	No Query Planning	0.812±0.010	0.908±0.009	0.885±0.009	17.2±0.8
	Full MAFA	**0.837±0.009***	**0.927±0.009***	**0.910±0.009***	20.5±0.9
CLINIC-150	1 Agent	0.838±0.004	0.930±0.002	0.900±0.002	12.1±0.2
	4 Agents	0.879±0.003	0.960±0.002	0.950±0.002	**12.0±0.2**
	No Query Planning	0.892±0.002	0.972±0.002	0.962±0.002	14.8±0.6
	Full MAFA	**0.901±0.003***	**0.980±0.002***	**0.970±0.002***	19.1±0.3

Table 2: Performance comparison across datasets (mean ± std over 10 runs). Bold indicates best performance. * denotes statistical significance ($p < 0.01$) compared to 4 Agents baseline using paired t-test.

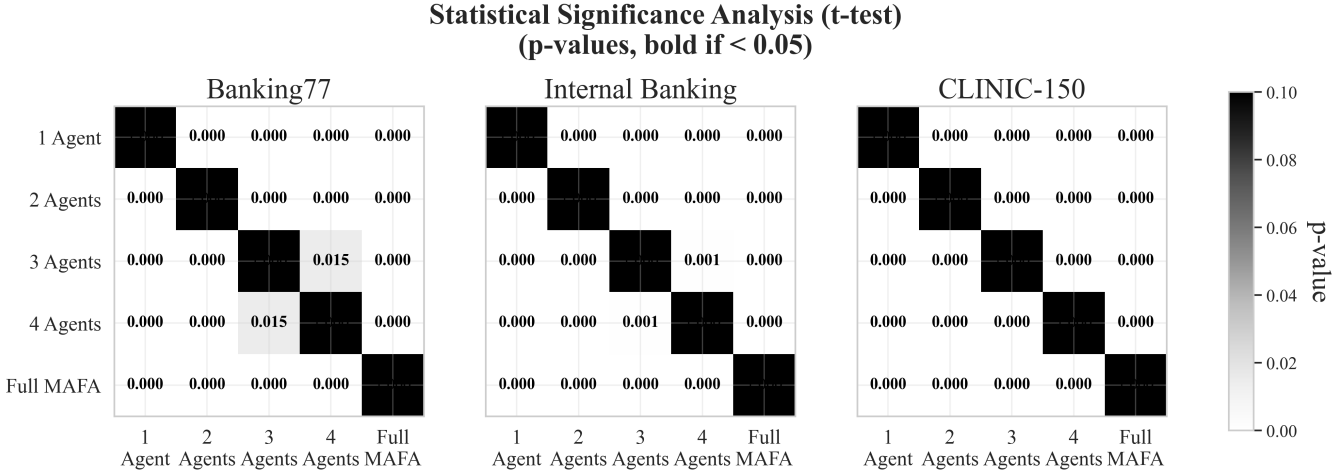

Figure 4: Statistical significance heatmap showing p-values for pairwise comparisons. Values below 0.05 (shown in bold) indicate statistically significant differences.

Configuration	Top-1 Accuracy	Δ
Full MAFA	**0.837±0.009**	–
Without Query Planning	0.812±0.010	-2.5%
Without Judge Agent	0.780±0.011	-5.7%
Without Embedding Agents	0.795±0.010	-4.2%
Without Few-shot Diversity	0.809±0.009	-2.8%
4 Agents Only	0.780±0.011	-5.7%
Single Agent Only	0.699±0.009	-13.8%

Table 3: Ablation study on Internal Banking dataset showing component contributions

3. **Multiple intents** (27% of errors): Queries legitimately spanning multiple intents

5.9 Key Findings

Our experimental evaluation demonstrates that:

1. MAFA consistently outperforms single-agent baselines by 6.3-13.8% in top-1 accuracy across diverse datasets
2. Each architectural component contributes meaningfully
3. Production deployment validates business value with 5000+ hours saved annually and a 26X annotator efficiency gain
4. Performance gains are statistically significant and consistent across multiple independent runs
5. The framework generalizes well accross annotation configurations

6 Lessons Learned

6.1 Technical Insights

Structured JSON-based prompting significantly reduces error rates—our deployment showed hallucination rates dropping from 8.3% to 5.2% compared to free-form Chain-of-Thought reasoning, with the schema acting as a guardrail that enforces explicit disambiguation steps. We also found that pure embedding approaches miss nuanced distinctions that LLM reasoning captures, and that distributing unique few-shot examples across agents improves ensemble performance more than sophisticated prompting alone.

Method	Top-1 Acc	Top-3 Acc	Top-5 Acc	MRR	NDCG@3	NDCG@5
BM25	0.120	0.145	0.210	0.382	0.401	0.431
Embedding-Only	0.185	0.210	0.465	0.545	0.668	0.692
Single-Agent (No Emb)	0.215	0.365	0.685	0.671	0.691	0.713
Single-Agent (Emb)	0.255	0.395	0.715	0.707	0.728	0.748
Single-Agent w/ Ans (No Emb)	0.220	0.450	0.690	0.712	0.703	0.713
Single-Agent w/ Ans (Emb)	0.270	0.480	0.730	0.722	0.743	0.763
MAFA	**0.355**	**0.625**	**0.865**	**0.790**	**0.802**	**0.815**

Table 4: Overall Performance Comparison on Bank dataset

Operational Metrics	Value
Total Utterances Processed	1,169,000
Historical Backlog Eliminated	1,067,400
Daily Processing Volume	8,000
Peak Hourly Throughput	3,500
System Uptime	99.7%
Quality Metrics	
Human Agreement Rate	86%
Previous Manual Agreement	72%
Annotation Consistency	+14%
Customer Query Resolution Rate	+8%
Model Performance Gain	+11%
Efficiency Gains	
Annotation Hours Saved (Annual)	5,187
Average Query Processing Time	0.42s
Previous Manual Time	11.1s
Efficiency Gain	26.4x

Table 5: Business impact metrics from production deployment (Jan-Oct 2024)

6.2 Operational Considerations

Successful deployment requires attention to three key factors. First, human-AI collaboration remains essential—the system achieves best results when integrated seamlessly with existing workflows rather than wholesale replacement. Second, confidence calibration plays a critical role: clear confidence thresholds enable appropriate routing between automated and manual processing, ensuring that ambiguous cases receive human oversight. Finally, continuous monitoring is necessary for production systems, requiring real-time performance tracking and rapid intervention capabilities to maintain reliability at scale.

6.3 Future Directions

Promising enhancements include active learning integration using low-confidence predictions to guide data selection, extending the framework to handle multi-intent utterances, and automated configuration learning to optimize annotation settings directly from data.

7 Conclusion

The deployment of MAFA at JPMorgan Chase represents a fundamental shift in how large institutions can leverage multi-agent AI systems for production-critical tasks. Over 3 months of operation, the system has processed $\sim$1.2 million utterances, eliminated a 1 million-query backlog, and saved 5,000+ hours of manual annotation effort.

Beyond raw metrics, MAFA's success demonstrates three critical insights for enterprise AI deployment. **First**, multi-agent architectures provide inherent resilience that monolithic models lack. When GPT-4 availability dropped during peak usage, our system gracefully degraded to three agents while maintaining 79% accuracy—sufficient for continued operation rather than complete failure. **Second**, the configurability of our framework proved essential for adoption. Different business units adapted MAFA for intents, products, and FAQ mapping without engineering involvement. This flexibility transformed a point solution into an enterprise platform now handling 8 distinct annotation tasks. **Third**, transparent confidence scoring enabled a cultural shift from "AI as replacement" to "AI as collaborator." Human annotators, initially skeptical, became advocates when they saw the system accurately flagging its own uncertainties. Job satisfaction increased as annotators shifted from repetitive labeling to handling complex, interesting edge cases; the 8% of queries where human judgment remains irreplaceable.

Looking forward, MAFA's deployment reveals both the promise and pragmatism required for enterprise AI. The system's 5,000+ hour annual value comes not from revolutionary algorithms but from thoughtful integration of existing techniques, careful attention to operational constraints, and relentless focus on user trust. As organizations worldwide grapple with annotation bottlenecks threatening their AI initiatives, MAFA provides a blueprint: start with clear business metrics, design for failure from day one, and remember that in production, 85% accuracy you can trust beats 95% accuracy you cannot.

With conversational AI becoming central to customer experience, annotation demands will only grow. MAFA shows that this challenge is solvable, not through moonshot research but through systematic engineering, empirical iteration, and balancing automation with human expertise; the foundation for robust, enterprise-ready AI.

Ethical Statement

The research reported in this paper reflects the independent work and opinions of the authors. It is not representative of the views or official policies of JPMorgan Chase.

References

Casanueva, I.; Temčinas, T.; Gerz, D.; Henderson, M.; and Vulić, I. 2020. Efficient Intent Detection with Dual Sentence Encoders. In Wen, T.-H.; Celikyilmaz, A.; Yu, Z.; Papangelis, A.; Eric, M.; Kumar, A.; Casanueva, I.; and Shah, R., eds., *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 38–45. Online: Association for Computational Linguistics.

Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J. B.; and Mordatch, I. 2023. Improving Factuality and Reasoning in Language Models through Multiagent Debate. *arXiv*.

Gilardi, F.; Alizadeh, M.; and Kubli, M. 2023. ChatGPT Outperforms Crowd Workers for Text-Annotation Tasks. In *Proceedings of the National Academy of Sciences*, volume 120.

He, X.; Lin, Z.; Gong, Y.; Jin, A.; Zhang, H.; Lin, C.; Jiao, J.; Yiu, S. M.; Duan, N.; and Chen, W. 2023. AnnoLLM: Making Large Language Models to Be Better Crowdsourced Annotators. *arXiv preprint arXiv:2303.16854*.

Hegazy, M. 2024. Diversity of Thought Elicits Stronger Reasoning Capabilities in Multi-Agent Debate Frameworks. *J Robot Auto Res*, 5(3): 01–10.

Karov, B.; Zohar, D.; and Marcovitz, Y. 2025. Attentive Reasoning Queries: A Systematic Method for Optimizing Instruction-Following in Large Language Models. arXiv:2503.03669.

Kusupati, A.; Bhatt, G.; Rege, A.; Wallingford, M.; Sinha, A.; Ramanujan, V.; Howard-Snyder, W.; Chen, K.; Kakade, S.; Jain, P.; and Farhadi, A. 2022. Matryoshka representation learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088.

Larson, S.; Mahendran, A.; Peper, J. J.; Clarke, C.; Lee, A.; Hill, P.; Kummerfeld, J. K.; Leach, K.; Laurenzano, M. A.; Tang, L.; and Mars, J. 2019. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1311–1316. Hong Kong, China: Association for Computational Linguistics.

Liu, N. F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; and Liang, P. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173.

OpenAI; :; Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; and Others. 2024. GPT-4o System Card. arXiv:2410.21276.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22.

Ratner, A.; Bach, S. H.; Ehrenberg, H.; Fries, J.; Wu, S.; and Ré, C. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. *Proceedings of the VLDB Endowment*, 11(3): 269–282.

Settles, B. 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Wang, J.; Wang, J.; Athiwaratkun, B.; Zhang, C.; and Zou, J. 2024a. Mixture-of-Agents Enhances Large Language Model Capabilities. *arXiv preprint arXiv:2406.04692*.

Wang, S.; Liu, Y.; Xu, Y.; Zhu, C.; and Zeng, M. 2021. Want To Reduce Labeling Cost? GPT-3 Can Help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 4195–4205. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Wang, X.; Kim, H.; Rahman, S.; Mitra, K.; and Miao, Z. 2024b. Human-LLM Collaborative Annotation Through Effective Verification of LLM Labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24. New York, NY, USA: Association for Computing Machinery. ISBN 9798400703300.