Insight Agents: An LLM-Based Multi-Agent System for Data Insights

Jincheng Bai
Amazon
Seattle, USA
baijin@amazon.com

Zhenyu Zhang
Amazon
Seattle, USA
zhenyuzh@amazon.com

Jennifer Zhang
Amazon
Seattle, USA
yezhan@amazon.com

Jason Zhu
Amazon
Seattle, USA
jasonzhu@amazon.com

Abstract

Today, E-commerce sellers face several key challenges, including difficulties in discovering and effectively utilizing available programs and tools, and struggling to understand and utilize rich data from various tools. We therefore aim to develop Insight Agents (IA), a conversational multi-agent Data Insight system, to provide E-commerce sellers with personalized data and business insights through automated information retrieval. Our hypothesis is that IA will serve as a force multiplier for sellers, thereby driving incremental seller adoption by reducing the effort required and increase speed at which sellers make good business decisions. In this paper, we introduce this new LLM-backed end-to-end agentic workflow designed for comprehensive coverage, high accuracy, and low latency. It features a hierarchical multi-agent structure, consisting of manager agent and two worker agents: data presentation and insight generation, for efficient information retrieval and problem-solving. We design a simple yet effective ML solution for manager agent that combines Out-of-Domain (OOD) detection using a lightweight encoder-decoder model and agent routing through a BERT-based classifier, optimizing both accuracy and latency. Within the two worker agents, a strategic planning is designed for API-based data model that breaks down queries into granular components to generate more accurate responses, and domain knowledge is dynamically injected to to enhance the insight generator. IA has been launched for Amazon sellers in US, which has achieved high accuracy of 89.5% based on human evaluation, with latency of P90 below 15s.

CCS Concepts

• **Information systems → Language models**; **Retrieval tasks and goals**.

Keywords

LLM, Agentic Workflow, RAG, Information Retrieval

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3731959

ACM Reference Format:
Jincheng Bai, Zhenyu Zhang, Jennifer Zhang, and Jason Zhu. 2025. Insight Agents: An LLM-Based Multi-Agent System for Data Insights. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3726302.3731959

1 Introduction

Running a small business involves numerous challenges, from product design and manufacturing to marketing and advertising. As an E-commerce company, we support sellers with powerful tools that leverage advanced machine learning to optimize pricing, forecast demand, manage inventory, identify market opportunities and many more. Although we provide sellers with rich tools to manage the facets of their business, it can be a challenge for some sellers to find the right tool, and leverage insights effectively. To remove these barriers, we develop Insight Agents (IA), which is an LLM-based [22] multi-agent conversational assistant, providing personalized data and business insight through automated information retrieval. It aims to reduce sellers' cognitive load and unlock their potential to grow business.

Specifically, IA enables seller to talk to their data through two main types of requests: (1) Descriptive Analytics that presents data according to the specified query. Examples include "what were my sales and traffic for the top 10 products last month", and "how does my monthly sales change year over year". (2) Diagnostic Analysis encompassing summarization, benchmarking, and other analytical techniques. Examples include "how does my business perform", "how is my business doing with respect to my benchmarks".

Building a reliable and helpful IA system is challenging, as it requires simultaneous consideration of coverage, accuracy and latency. To circumvent these challenges, we propose an end-to-end agentic workflow that employs a hierarchical manager-worker multi-agent structure to optimize data retrieval and question answering, which follows the general multi-agent concept [6, 21] utilizing different resolution paths tailored to the query type. The **manager agent** mainly consists of Out-of-Domain (OOD) detection and branch routing. We build specialized lightweight models for OOD (an encoder-decoder based detector) and branch routing to resolution path to optimize latency (a BERT-based classifier). We then divide the solution space into two worker agent-based resolution paths, **data presenter agent** and **insight generator**

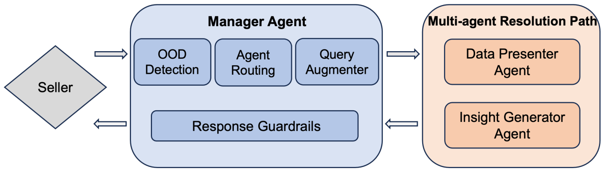

Figure 1: Overall IA Architecture. It illustrates the overall hierarchical structure, consisting of a manager agent overseeing two subordinate worker agents: data presenter agent and insight generator agent.

agent, to facilitate problem-solving. Essentially, IA is built upon the Retrieval-Augmented Generation (RAG) [8] framework, leveraging seller data in a tabular format through a robust and strategic API-based data model. The system decomposes queries into appropriate grains based on company's internal data API availability in a divide-and-conquer manner, with automated tabular data retrieval and aggregation facilitated through task decomposition, planning, and API/function selection. This process, powered by LLM, is analogous to LLM-based task planning [10, 14] and tool selection [16, 19]. In addition, we inject domain knowledge dynamically based on query to make insight generator domain-aware. Overall, we implement the manager-worker multi-agent agentic workflow with orchestration architecture to streamline the entire process, encompassing query understanding, information retrieval and answer generation. IA has been launched for Amazon sellers in US, which has achieved accuracy of 89.5% based on human evaluation, with latency of P90 below 15s.

2 Methodology

2.1 Architecture Design

Overall IA Architecture. The high-level hierarchical manager-worker multi-agent architecture of IA is illustrated in Figure 1. Upon receiving a seller's query, the manager agent first checks its eligibility against the scope of data insight via Out-of-Domain (OOD) component. It also includes the agent routing component to select the appropriate solver, and the query augmenter to de-ambiguate the query based on its best knowledge. We design two distinct agent-based resolution paths, i.e., data presenter and insight generator agent which are separable in terms of the solution they provide. Before returning response to sellers, IA applies guardrail to prevent response that contains issues such as PII leakage, toxic message, etc, from exposing to the seller as the post-processing step.

Design for Data Presenter and Insight Generator. Figure 2 illustrates the low-level architecture design of the IA, specifically elucidating the details of the two agents in the resolution paths, which will be explained in section 2.3.

Implementation Considerations. One bottleneck for LLM-based agents is the latency. To overcome this, there are two considerations to address it: model-based components and parallelization. Specifically, we develop specialized model to avoid unnecessary LLM calls while ensuring high performance and only apply LLM for components that involve data fusion and generation. Regarding parallelization, OOD detection and agent routing are performed simultaneously, with both the data presenter and insight generator branches being initiated concurrently with early termination. The parallelization approach trades off increased infrastructure and computation costs for latency reduction.

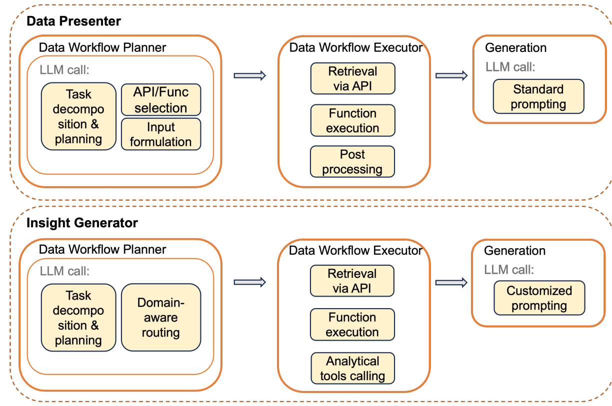

Figure 2: Architecture design for data presenter and insight generator.

2.2 Manager Agent

Out-of-Domain (OOD) Detection. The out-of-domain (OOD) detection within manager agent serves as the first gating after receiving the query. The component understands seller intent to determine whether it is out of scope for the data insight agent to answer. The in-scope is defined as at least part of the question can be answered based on the available data. We create a high-precision classifier that favors precision over recall when filtering data insight requests, to act as an initial screening layer without running the full analysis pipeline. While some invalid requests may not be filtered out (false negatives), they can still be caught later in the data workflow planing stage. Specifically, we implemented Auto-encoder (AE) based OOD detection, similar to [20] to depict the boundary of in-scope problem space. In particular, the Auto-encoder with one hidden layer is trained with embedding from sentence transformer [18]. Denote the input embedding vector as X, and the encoder transforms it to a hidden representation H as $H = \sigma(W_1 X + b_1)$ and followed by the decoder, $\hat{X} = \sigma(W_2 H + b_2)$ with weights W_1, b_1, W_2, b_2. The AE is trained to minimize the reconstruction error $r = ||X - \hat{X}||$. We train the AE on the the set of in-domain questions denoted as $\mathcal{X}_{id}$, and the threshold for OOD is determined by

$$\mu_{id} + \lambda * \sigma_{id} \tag{1}$$

where μ_{id} and σ_{id} denotes the mean and standard deviation of the reconstruction loss r on $\mathcal{X}_{id}$. The hyper-parameter λ controls the precision v.s. recall in the OOD detection.

Agent Router. The agent router solves a classification problem by categorizing input queries between data presenter and insight generator. To strike a balance between latency and accuracy, a lightweight transformer-based model could be employed, ensuring responsive performance while maintaining robust classification capabilities. In particular, we fine-tuned a lightweight BERT [5] based model (33M parameters) on super-sampled data, encompassing presenter and insight generator questions with variations.

Query Augmenter. It clarifies, rewrites, and expands queries to reduce ambiguity. In particular, the ambiguity regarding data insight related questions often centers around the time range specified. For example, question "*What were my sales for the last week?*" lacks clarity like what the current date is, and the LLMs might also have difficulty accurately interpreting time ranges like "last week". In this step, contextual information like today's date, the start and end dates of the current week are dynamically injected with specific instructions like "week" referring to the calendar week. This step augments the prompt for LLM call in the subsequent step.

2.3 Data Presenter and Insight Generator Agents

We examine the shared components of Data Presenter and Insight Generator Agents, highlighting their key differences.

2.3.1 Data Workflow Planner: a Robust Data Model. Our modeling framework relies on retrieval-augmented generation (RAG) technique [8], as the insight related responses are grounded on seller data. As specific to IA, the external resources are stored in tabular form, which necessitates the development of tabular retrieval method different from retrieving from unstructured text data. To ensure accuracy of data retrieval and aggregation, we propose a robust data model to retrieve data that leverages company's internal data APIs, decomposes the query into solvable steps based on API availabilities, following a Divide and Conquer manner. It specifically uses APIs to retrieve data, which naturally imposes structure and constraints but yields higher accuracy. This represents a trade-off between flexibility and precision when compared to text-to-SQL solutions [7, 11, 13, 15], which additionally require the effort of creating and maintaining a relational database. Besides data APIs, leveraging external calculation tools for data transformation is also crucial as calculation errors remain a common challenge for LLMs [1, 12]. The whole process is similar to tool learning and usage by LLM [17], with few-shot examples and tool metadata stored in memory. Workflow planner encompasses task decomposition and planning, along with the selection of appropriate APIs/functions and payload generation with slot filling. An end-to-end illustration for data workflow planner can be found in Figure 3.

Data-based Out-of-Scope Detection. The LLM performs secondary out-of-scope detection by comparing incoming queries against provided dataset metadata. An explicit "out" option is provided to LLM to prevent hallucination when queries exceed available data boundaries.

Task Decomposition and Planning. LLM-based task decomposition and planning [10, 14] leverages chain of thought (CoT) [23] to ensure comprehensive instruction adherence and few-shot learning [3] to guide the process. The data presenter enables customized data retrieval and aggregation through query decomposition into executable steps, by providing detailed API/function descriptions to LLM. The insight generator decomposes questions into domain-specific categories (such as performance, benchmarking, recommendation, etc) if necessary, and then selects predefined domain-aware resolution paths.

API/function Selection with Payload Generation. The API and function selection is akin to LLM-based tool selection [16, 19]. The API/function name, description and column name are provided in the prompt to facilitate the process. This step essentially resembles the schema linking [9], i.e. table & columns linking, where LLM creates the alignment between the entity references in the given query and the schema tables or columns. Subsequent to the API/function selection process, the next step involves payload/input generation with slot filling, where the required input parameters or arguments for the chosen tools or APIs are properly populated to facilitate their effective execution. In essence, the whole process is similar to code generation such as text-to-SQL, yet it is more robust in the sense that it is less prone to syntax errors and hallucinations (especially columns) associated with text-to-SQL.

Domain-aware Routing for Insight Generator. The insight generator also employs a few-shot learning based LLM classifier for domain-specific branch routing. The predetermined domain-aware resolution paths mainly contain (1) the associated analytical analysis that may include data aggregation, time series based seasonal & trend analysis, as well as benchmark analysis among other techniques. (2) the domain-aware knowledge, prompt template and few-shot examples for the final insight generation.

2.3.2 Data Workflow Executor and Generation. Data Workflow Executor component carries out data retrieval and data aggregation/transformation operations, adhering to the instructions outlined by the Data Workflow Planner. Additionally, data post processing tasks, such as reformatting the data, column renaming and semantic matching-based column filtering are also performed.

The generation components mainly leverages in-context learning (ICL), with few-shot learning approach [3] and CoT [23], to generate insights. The generation process for the data presenter is straightforward, with few-shot examples guiding the desired format of the response. Regarding the insight generator, domain experts provide specific domain knowledge and few-shot examples tailored to different data sources for LLM to follow.

3 Experiments

3.1 Experimental Setup

3.1.1 Datasets. To train OOD detection and agent routing models, we collected 301 commonly asked questions, with 178 in-domain and 123 out-of-domain respectively. The in-domain questions are further divided into 120 queries for the data presenter and 59 for the insight generator. To facilitate finetuning of the lightweight BERT model with a balanced dataset, the raw data presenter and insight generator questions are further augmented by LLM to be super-sampled to 300 questions each, introducing variations to expand the dataset. To evaluate IA end-to-end, a benchmarking dataset consisting of 100 carefully selected popular questions with ground truth are constructed.

3.1.2 Setup. The LLM employed in the experiments is based on "anthropic.claude-3-sonnet-20240229-v1:0" [2] via Amazon Bedrock. Regarding the configuration of the OOD model, in our implementation the hyperparameter λ is set as 4, and the dimension the hidden layer is 64. The base BERT model is "bge-small-en-v1.5" (33M parameters) [4].

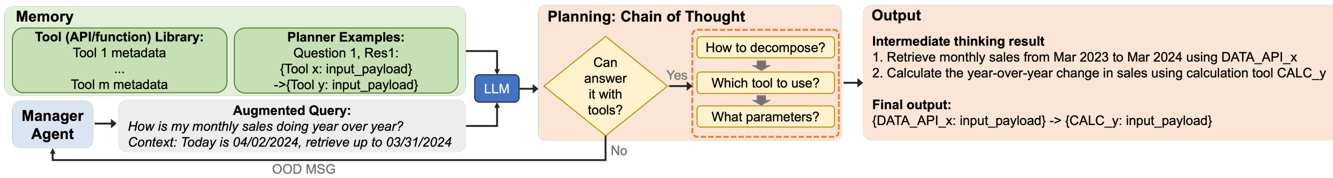

Figure 3: Illustration for data workflow planner. Use data presenter as an example.

Table 1: OOD Detection Performance

Model	Precision	Recall	Running Time (second)
Auto-encoder	0.969	0.721	0.009
LLM-based few-shot	0.616	0.971	1.665

3.1.3 Metrics. The performance of OOD model and agent routing model are gauged via metrics such as precision, recall and accuracy. Since there are multiple combinations of data retrieval/aggregation, we evaluate the retrieval performance together with end-to-end model performance, where the IA response is evaluated by human auditors on the benchmarking datasets. The response of IA are evaluated from the following three quantitative dimensions:

Relevance: Assess whether the key points and insights provided in the response directly address the question, which is defined as *relevance = #_key_words from the question that were addressed in the response / total #_key_words in the question.*

Correctness: Verify the accuracy and reliability of the data and information presented in the response (similar to the notion of precision). *correctness = #_correct_insights in response / total #_insights in response.*

Completeness: Evaluate whether the response covers all the necessary data points or metrics that the user is inquiring about (akin to the concept of recall). *completeness = #_required_insights in response / total #_required_insights.*

The question-level accuracy is defined as **Question-level Accuracy**: the percentage of questions that have correctness, completeness, relevancy of more than 0.8.

3.2 Experimental Results

3.2.1 OOD Detection. As shown in table 1, AE-based OOD only takes less than 0.01s over each testing sample which significantly beats record compared to the LLM-based method. Meanwhile, it outperforms LLM significantly in terms of precision. As a further enhancement, the in-domain set (training set) could be enlarged to improve recall.

3.2.2 Branch Routing. The results of branch routing is in table 2. Out of 178 in-domain samples, the model achieves classification accuracy of 0.83 with 0.3s latency for each routing case, compared with 0.60 accuracy in LLM-based classifier with >2s latency.

3.2.3 Human Evaluation. For end-to-end IA response evaluation, the benchmarking dataset of 100 questions is sent to human auditors for evaluation providing the rubrics and ground truth. Out of 100 questions, 57 questions are in-scope, and the summary for the quantitative measures can be found in the table 3.

Table 2: Branch Routing Performance

Model	Accuracy	Running Time (second)
Finetuned BERT	0.83	0.31
LLM-based few-shot	0.60	2.14

Table 3: Summary of evaluation metrics

Metric	Avg	Std	Min	Max	Median	Samples
Relevancy	0.977	0.102	0.5	1	1	57
Correctness	0.958	0.125	0.455	1	1	57
Completeness	0.993	0.045	0.714	1	1	57

Table 4: Question-level accuracy

Question-level Accuracy	Count	Percentage
False	6	10.5
True	51	89.5

Meanwhile, the question-level accuracy is summarized in table 4. The overall accuracy is high as 89.5%. The end-to-end P90 latency is at 13.56s.

4 Conclusions

In this paper, we introduced Insight Agents (IA), a hierarchical multi-agent system leveraging LLMs to provide personalized, actionable insights for e-commerce sellers. Our system significantly reduces cognitive load, achieving 89.5% accuracy with sub-15-second latency. Looking ahead, we aim to scale IA to cover more use cases and integrate automated evaluation techniques for enhanced performance. Ultimately, IA sets the stage for future AI-driven decision supporting systems, transforming data interaction and driving impactful outcomes in e-commerce and beyond.

Appendix: sample outputs

Q: "What were the sales for my top 10 products last month?", A: "Your top 10 products by Sales for August 2024 (2024-08-01 2024-08-31) were: 1. Product1: $30,000, 2. Product2: $20,000, ..."
Q: "How is my business performing?" , A: "In Feb 2024, your sales was $10K (-$6K or -50% monthly YoY). Number of items sold were 200 (-50 or -20% YoY). Average selling price came in at $50 (-$20 or -30% YoY). Traffic was 30K views (+50% YoY). Conversion rates were 0.50% (-50 bps YoY). Overall business insights: ..."

Presenter Biography

Jincheng Bai is an Applied Scientist at Amazon, focusing on Gen AI and Agentic LLM projects. He received his PhD in Statistics from Purdue University in US. His research mainly focused on deep learning and natural language processing, and he has published at top-tier conferences including NeurIPS.

References

[1] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large Language Models for Mathematical Reasoning: Progresses and Challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*. 225–237.

[2] Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 1877 – 1901.

[4] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv preprint arXiv:2309.07597* (2023).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2016. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019*. 4171–4186.

[6] Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, Katsushi Ikeuchi, Hoi Vo, Li Fei-Fei, and Jianfeng Gao. 2024. Agent AI: Surveying the Horizons of Multimodal Interaction. *arXiv preprint arXiv:2401.03568* (2024).

[7] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, and et al. 2023. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *arXiv preprint arXiv:2308.15363* (2023).

[8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, and et al. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* (2024).

[9] George Katsogiannis-Meimarakis and Georgia Koutrika. 2023. A survey on deep learning approaches for text-to-SQL. *The VLDB Journal* 32 (2023), 905–936.

[10] Chuanhao Li, Runhan Yang, Tiankai Li, Milad Bafarassat, Kourosh Sharifi, Dirk Bergemann, and Zhuoran Yang. 2024. Stride: A tool-assisted llm agent framework for strategic and interactive decision-making. *arXiv preprint arXiv:2405.16376* (2024).

[11] Haoyang Li, Jing Zhang, Cuiping Li, and Hong Chen. 2023. RESDSQL: Decoupling Schema Linking and Skeleton Parsing for Text-to-SQL. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. 13067–13075.

[12] Xiaoyuan Li, Wenjie Wang, Moxin Li, Junrong Guo, Yang Zhang, and Fuli Feng. 2024. Evaluating Mathematical Reasoning of Large Language Models: A Focus on Error Identification and Correction. In *Findings of the Association for Computational Linguistics: ACL 2024*. 11316–11360.

[13] Aiwei Liu, Xuming Hu, Lijie Wen, and Philip Yu. 2023. A comprehensive evaluation of ChatGPT's zero-shot Text-to-SQL capability. *arXiv preprint arXiv:2303.13547* (2023).

[14] Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. 2023. Art: Automatic multistep reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014* (2023).

[15] Mohammadreza Pourreza and Davood Rafiei. 2023. DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 36339 – 36348.

[16] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and et al. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *In Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

[17] Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2024. Tool Learning with Large Language Models: A Survey. *arXiv preprint arXiv:2405.17935* (2024).

[18] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3982–3992.

[19] Yifan Song, Weimin Xiong, Dawei Zhu, Cheng Li, Ke Wang, Ye Tian, and Sujian Li. 2023. Restgpt: Connecting large language models with realworld applications via restful apis. *arXiv preprint arXiv:2306.06624* (2023).

[20] Hasan Torabi, Seyedeh Leili Mirtaheri, and Sergio Greco. 2023. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity* 6, 1 (2023).

[21] Lei Wang, Ma Chen, Xueyang Feng, Zeyu Zhang, and et al. 2024. A Survey on Large Language Model based Autonomous Agents. *Frontiers of Computer Science* 18 (2024), 1–42.

[22] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and et al. 2022. Emergent abilities of large language models. In *Transactions on Machine Learning Research (2022)*.

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 24824 – 24837.