# Boosting not so Large Language Models by using Knowledge Graphs and Reinforcement Learning

**William Jones Beckhauser**[1], **Renato Fileto**[1]

[1]Department of Informatics and Statistics (INE)
Federal University of Santa Catarina (UFSC), Florianópolis - Brazil

{beckhauserwilliam@gmail.com, r.fileto@ufsc.br}

***Abstract.*** *Ensuring the viability of large language models (LLMs) in situations requiring data privacy with limited on-premise resources is a significant current challenge. This work investigates how to tackle this challenge using knowledge graphs (KGs) and reinforcement learning (RL) to enhance minor LLMs by reducing non-factual responses and response gaps. We evaluated variations of GPT (4o, 4, and 3.5), Llama2 (7b, 13b, and 70b), and Llama3 (8b and 70b) for multi-label classification and information extraction, with or without KG and RL, and also fine-tuned a BERT model. Llama3 8b combined with KG and RL outperformed all other LLM models, and the fine-tuned BERT model too.*

## 1. Introduction

Large language models (LLMs) such as GPT [Liu et al. 2023], Llama [Gao et al. 2023], and Gemini [Team et al. 2023] are increasing their parameter count with each new release, for performance gains [Xue et al. 2024]. Nevertheless, this technology, usually available in the clouds of large private corporations, remains out of reach for many companies and projects that need to operate on local servers [Yao et al. 2024], due to high costs and regulations like the General Data Protection Law (LGPD) [Erickson 2018]. These enterprises could rely on open-source models with many parameters, but their computational requirements are too high to run on-premises [Alizadeh et al. 2023].

Nowadays, there is a subtle research movement towards smaller open-source LLMs [Shridhar et al. 2023, Shen et al. 2024], and an intense pursuit of optimization strategies. A promising direction is Retrievable Augmented Generation (RAG) using a Knowledge Graph (KG) to add relevant formal knowledge to LLMs [Pan et al. 2023]. This approach has been tested in various tasks, including fake news detection [Liu et al. 2024], text classification [Shi et al. 2023], and refined node classification in citation graphs and networks [Bruno et al. 2023, He et al. 2023]. In the biomedical domain, these solutions have been applied in recommendation systems and drug-gene interaction studies [Xu et al. 2024, Wang et al. 2023], as well as in recruiting for clinical studies [Guan et al. 2023]. However, there are still few concrete examples demonstrating consistent performance gains by using approaches like Graph-RAG [Pan et al. 2023] in typical machine learning tasks, such as multi-label classification or information extraction, especially when using open-source LLMs.

This article contributes to filling this research gap by evaluating the synergism of KGs, reinforcement learning (RL) and LLMs. We compare the performance of relatively small LLMs, like Llama2 (7b and 13b) and Llama3 (8b), with that of larger LLMs

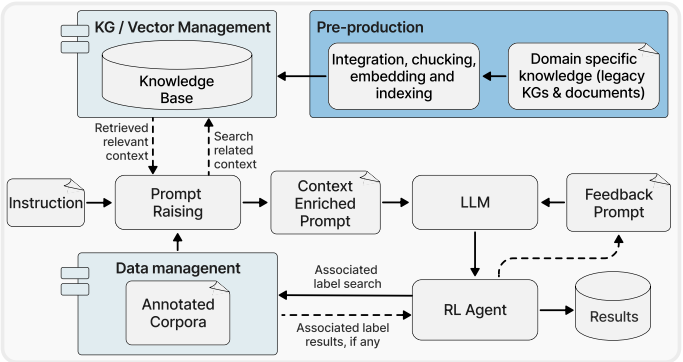
like GPT (4, 4o, and 3.5), Llama2 (70b), and Llama3 (70b), each one alone and combined with the use of KGs and RL, in two tasks: (i) multi-label classification of reviews posted by users of a food delivery app in multiple languages and their translations into English, and (ii) information extraction from invoices of different types of backhoes. We propose and evaluate alternative approaches for exploiting domain specific KGs to enrich LLM prompts with relevant context. An RL agent validates responses, restricting them to predefined labels, when available, and providing feedback to the models. It randomly validates some LLM responses with their respective labels throughout the RL process. We also fine-tuned and evaluated a BERT model for performing the same multi-label classification, on the same datasets.

The main contributions of this article are: (i) a systematic evaluation of language models, considering each LLM alone and assisted by a KG and/or an RL agent; (ii) demonstrating the superiority of smaller, open-source models, like Llama3 8b, when combined with KGs; (iii) showcasing the feasibility of feedback systems for language models; and (iv) applying LLMs combined with KGs and RL in unexplored fields.

## 2. Proposed Approach

Figure 1 shows the architecture of our integrated Graph-RAG and RL system for LLMs, designed to optimize responses in classification and information extraction tasks. The process starts with a instruction sent to the Prompt Raising module, supplemented by data from annotated corpora (e.g., a backhoe invoice). This module interacts with the KG/vector management component to search the Knowledge Base for relevant context by accessing the knowledge graph linked to the instruction. The retrieved context is then integrated into the prompt, which the LLM uses to generate a response. The RL Agent checks the LLM's output against available labels(train data). If inaccuracies arise, feedback is given to the LLM, and interaction results are stored in the Results database.

**Figure 1. Proposed process for using knowledge and RL to improve LLM results.**




### 2.1. Pre-production

In the pre-production phase, we focus on constructing KGs using domain-specific structured data sources. For example, in information extraction tests related to backhoe loaders, we use tables with product descriptions segmented into products, brands, and models. These are organized into a hierarchy of classes and subclasses, with connections like "Product" connected by "offered by" to "Brand," which in turn connects via "has"

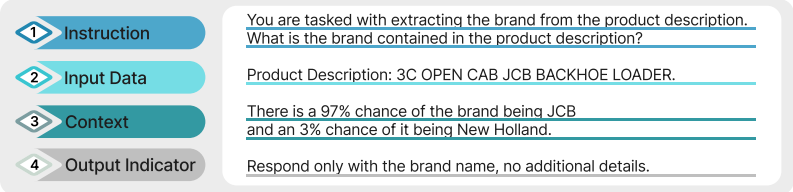
to “Model”. Once the structure is defined and validated, we generate embeddings for the classes, subclasses, and connections using the BGE model [Chen et al. 2024]. The graphs are then implemented in the Neo4j graph database, incorporating the generated embeddings. These KGs stored in Neo4j serve as our Knowledge Base.

## 2.2. Prompt Raising

In the "Prompt Raising" phase, the system processes three inputs: the instruction, textual descriptions on data management, and relevant information from the knowledge base. The third input is obtained via two methods: Graph-RAG Targeted, retrieving highly similar information, and Graph-RAG Comprehensive, gathering related classes and their interrelationships without filters.

Embeddings are generated from the inputs using the BGE model, the same one employed for knowledge graphs (KGs). Then, a similarity search compares these embeddings with the knowledge base using cosine similarity. The Graph-RAG Targeted method identifies records with a cosine similarity above 85%, while the Graph-RAG Comprehensive method retrieves all relevant classes, subclasses, and connections. For example, when processing "3C OPEN CAB JCB BACKHOE LOADER" in a brand-related instruction, Graph-RAG Targeted might indicate "97% probability for JCB and 3% for New Holland," while Graph-RAG Comprehensive would provide broader insights such as "The JCB brand includes models 3CX and 5CX" and "New Holland covers models B95C and B115C". Thus, as shown in Figure 2, the output of Prompt Raising consists of Instruction and Output Indicator. The Input Data represents a textual description from Data Management, and the Context, in this example, is Graph-RAG Targeted, which retrieved the data with the highest cosine similarities from our Knowledge Base.

**Figure 2. Graph-RAG-Enhanced Contextual Prompt for Information Extraction.**




## 2.3. LLM

We configure the LLM and invoke its API using an enriched prompt derived from the Prompt Raising stage. Key parameters, like temperature and output token count, are adjusted. Temperature controls prediction randomness, with lower values yielding more deterministic results and higher values increasing creativity. For classification and extraction tasks, we limit the output to fewer than 10 tokens. We employ models like Llama2 (7b, 13b, 70b) and Llama3 (8b, 70b) via the Deepinfra API, as well as GPT models (3.5, 4.0) via the OpenAI API. With the enriched prompt and optimal model settings, the API is called to perform extraction or classification. For example, for the product description “3C OPEN CAB JCB BACKHOE LOADER”, the expected response would be “JCB”.

## 2.4. RL Agent

The RL Agent processes the output of the LLM model by checking if there is a corresponding label in the database, as detailed in the enriched prompt. The annotated corpora

include a percentage of pre-labeled data randomly distributed, and each new LLM output is compared against these corpora. For example, if the LLM classifies a product description as "New Holland" for "JCB 3C OPEN CAB Backhoe Loader," the RL Agent searches the annotated corpora to check if there is a label. If "New Holland" is correct or if there is no existing label, the response is validated and stored; if incorrect, the agent provides feedback suggesting the correct label. This process is repeated up to five times to correct and reinforce the model's learning. For classification tasks with predefined labels, the RL Agent adopts a two-step validation process. First, it checks if the LLM's classification matches the predefined labels. If it doesn't match, the agent provides feedback to align the response with the established categories. In the second step, if the classification falls within the categories, the Agent validates it against the associated label (if any).

## 3. Application Scenarios

### 3.1. Multi-label Classification of Food Delivery Reviews

In our first scenario, we analyzed a dataset of around 4,000 customer reviews from a European food delivery app, ranging from 0 to 889 characters, available in [Beckhauser and Fileto 2024]. After removing duplicates and outliers, 3,451 reviews remained. Approximately 80% are in European Portuguese, with the rest in English, Spanish, Italian, and Catalan. Given the importance of English in LLM training, we created a parallel dataset by translating all reviews into English using Googletrans, with manual corrections for about 30 reviews. We then identified key terms for each label by removing stop-words in multiple languages using nltk and spaCy and extracting frequent words with the Counter library. For sentiment analysis, we used the SiEBERT model [Hartmann et al. 2023], which showed consistent performance, even when compared to GPT-4 [Krugmann and Hartmann 2024]. Sentiment analysis results and dataset details are summarized in Table 1.

We manually built a tree-like KG to categorize reviews, distinguishing between "Product" (item-related) and "Order" (delivery/service-related). Subcategories like "Quantity issue" and "Quality issue" under "Product," and "Delivery issue" and "Praise comment" under "Order" are further refined with specific keywords.

**Table 1. Dataset review distribution by class, subclass, and sentiment.**

| Class | Subclass | Description | #Reviews | % | Pos. | Neg. |
|---|---|---|---|---|---|---|
| Product | Quality issue | Issues with food preparation, taste, or hygiene. | 671 | 19.44 | 26% | 74% |
| Product | Quantity issue | Dissatisfaction with the amount or size of the portions served. | 605 | 17.53 | 28% | 72% |
| Order | Delivery issue | Problems related to delays, wrong deliveries or missing items. | 1196 | 34.66 | 26% | 74% |
| Order | Praise comment | Positive comments about the quality of the service or product. | 979 | 28.37 | 98% | 2% |

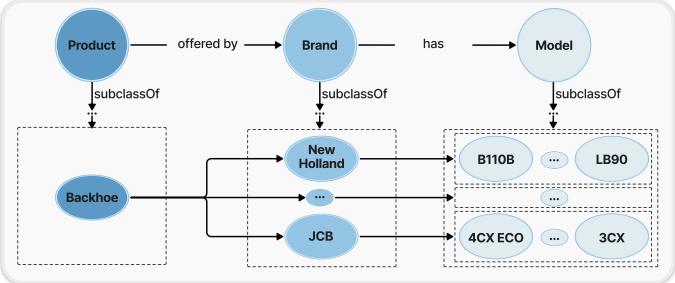
### 3.2. Information Extraction from Invoices

Our second application scenario involves a dataset of approximately 17,000 work machine purchase invoice descriptions, including the Mercosur Common Nomenclature (NCM)

and unit item values, provided by (blind review). The invoice descriptions range from 16 to 120 characters. Initially, we filtered the dataset using the NCM code, focusing on the first four characters, specifically "8429," which covers bulldozers, graders, excavators, and similar machinery. We then applied a keyword dictionary to identify relevant terms. Backhoe loaders appeared most frequently, with around 1,100 descriptions, becoming the primary focus of our experiments. This dataset lacked initial classifications, containing only raw invoice descriptions. To facilitate future model validation, we manually categorized the data into predefined classes such as brand, model, and specifications. A dictionary comprising brands, models, keywords, orthographic variations, acronyms, and abbreviations was used, considering possible typographical errors. Fields not covered by the dictionaries were manually completed, ensuring thorough validation of LLM outputs.

**KGs for Backhoe Invoices.** Figure 3 shows an extract of an ontology in KG format, centered on heavy machinery. It depicts the 'Product' concept, with 'Backhoe' as a subclass, linked to 16 brands via the "offered by" relation. Brands like 'New Holland' and 'JCB' are highlighted, each connected to specific models through the "has" relation. For instance, 'New Holland' includes models like 'B110B' and 'LB90,' while 'JCB' offers '4CX ECO' and '3CX.' In total, 68 models are represented.

**Figure 3. KG extraction with concepts and relations from heavy machinery.**




## 4. Experiments

In this section, we describe the experiments conducted for multi-label classification with customer reviews, subsection 4.1, and information extraction experiments using backhoe invoice data, subsection 4.2. All datasets and models were tested in various distinct scenarios: (1) classification or extraction using only the instruction and corpus, without providing enriched context to the LLM; (2) using only the RL Agent; (3) adding a comprehensive search in the KGs, which returns all classes, subclasses, relationships, and leafs as context; (4) using targeted context with similarity search above 85%, utilizing Graph-RAG; (5) using Graph-RAG Comprehensive with RL; (6) using Graph-RAG Targeted with RL. Additionally, for the multi-label classification experiments, we will conduct a test with embeddings and fine-tuning using BERT. A more comprehensive description of the experiments developed is available at GitHub[1].

### 4.1. Multilabel Classification of Customer Reviews

In this subsection, we present the experiments conducted for multi-label classification. The experiments are performed on two subsets of customer review data: the first contains

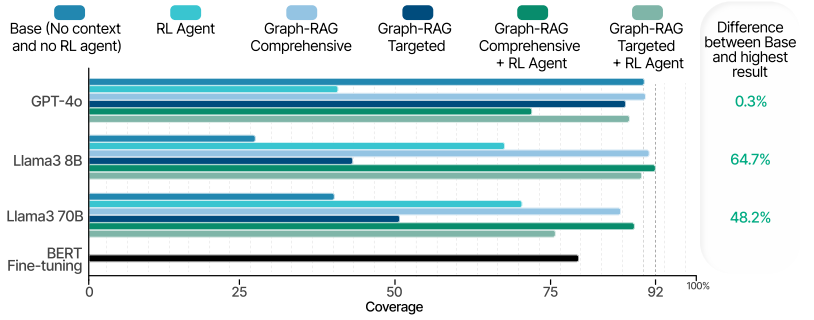
[1] https://github.com/WilliamBeckhauser/Boosting-not-so-LLM

reviews from customers in various languages, and the second comprises the same reviews translated into English. Each dataset includes 3,451 reviews. We randomly selected 300 reviews from each label for the agent to use as a validator during the classification process, resulting in 1,200 reviews used solely for reinforcement training on the model.

The BERT experiment tokenize reviews and split them into training and testing datasets at an 80/20 ratio. We use BERT to produce embeddings and a training function with an AdamW optimizer and a linear scheduler. To optimize hyperparameters, we set up an objective function in Optuna, adjusting the learning rate, weight decay, and epochs.

**English dataset:** In these experiments, the Llama3 8b model, when combined with Graph-RAG Comprehensive and an RL agent, achieved a 64.7% increase in coverage compared to the "Base" experiment, the highest among all models and scenarios (Figure 4). Without Graph-RAG Comprehensive and the agent, coverage dropped drastically to 27.2%. The Llama3 8b also excelled in precision (93.3%) and F1-Score (92%) under the same conditions.

**Figure 4. Multi-label classification of customer reviews in English.**



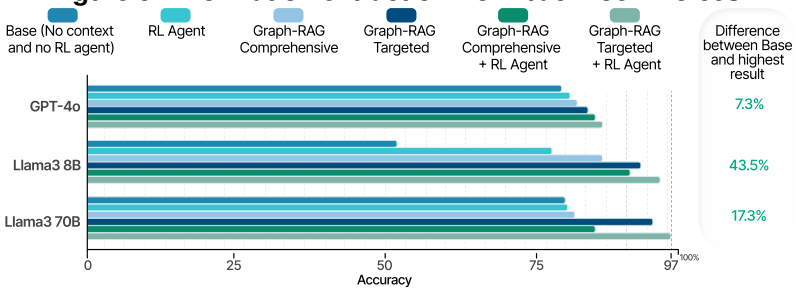

The GPT-4o model, when used solely with Graph-RAG Comprehensive, recorded 90.4% coverage, 89.1% precision, and an 89.1% F1-Score, outperforming its configuration with agents, where these metrics were 72%, 71.1%, and 71% respectively. GPT-3.5 showed moderate stability, with 81.8% coverage, 84.5% precision, and 80.6% F1-Score using Graph-RAG Comprehensive and agents. Without agents, these values only slightly declined to 81.3%, 84.1%, and 80%. The Llama2 variants underperformed, particularly the 13b version without agents. The Llama3 70b model improved in precision (92%) and F1-Score (91%) with RL agents but showed reduced performance without them. The BERT model achieved 79% precision and 76% F1-Score with general context and RL agent conditions, but still lagged behind the Llama3 models.

**Multilingual dataset:** The coverage improvements were more modest for GPT-4 and GPT-4o models, with only a 0.2% increase, but they retained high accuracy (around 88-89%). Notably, Llama3 70b showed strong results in both contexts, with 90.1% coverage in the multilingual setting and consistently high precision across datasets. However, in none of the scenarios did the multilingual dataset surpass the results achieved with the English dataset, highlighting a clear performance gap. Smaller models like Llama2 13b particularly struggled in both datasets, especially in multilingual tests where coverage remained low even with advanced techniques. The findings emphasize the superior adaptability of larger models like Llama3 and GPT-4 across languages, while smaller models struggle to maintain effectiveness without additional enhancements.

### 4.2. Information Extraction from Backhoe Invoices

Figure 5 shows that the Llama3 8b model, when operated with KGs and RL agents, displays a remarkable improvement in accuracy. Specifically, the accuracy increased from a baseline of 52.18% to 95.7% when using Graph-RAG Targeted and RL Agent, demonstrating an enhancement of 43.52%.

**Figure 5. Information extraction from backhoe invoices.**




The Llama3 8b model achieved the highest accuracy of 95.7% and the greatest accuracy improvement among the configurations, illustrating its strong synergy with KGs and RL agents. Conversely, without these tools, its accuracy substantially decreases to the baseline of 52.18%. For the Llama3 70b model, the highest accuracy reached was 97.21% with Graph-RAG Targeted and RL Agent, showing a slight accuracy increase from its baseline of 79.93%. This model also exhibited the highest consistency across different configurations. The GPT-4o model showed improvements as well, reaching an accuracy of 86.48% with Graph-RAG Comprehensive and RL Agent, which is an increase of approximately 7.25% over its baseline of 79.23%. These results highlight the significant impact of utilizing KGs and RL agents in enhancing the performance of machine learning models, especially in tasks that involve complex document analysis such as information extraction from backhoe invoices.

### 4.3. Discussion

This study aligns with the growing body of research exploring the potential of LLMs to address NLP challenges. Although these models are capable of handling a wide range of tasks without the need for specialized data, in more specific cases, they show significant limitations due to the lack of fine-tuning, especially in smaller versions. LLMs face substantial limitations in their reasoning abilities, particularly when dealing with tasks involving multiple languages. In these scenarios, current LLMs still do not outperform approaches that utilize RL, whether through techniques like Proximal Policy Optimization (PPO), Trust Region Policy Optimization (TRPO), or Deep Deterministic Policy Gradient (DDPG), which require deep model adjustments, making their application considerably costly, or through RL techniques that provide textual feedback, as explored in this work. Consequently, approaches like Graph-RAG or RL with textual feedback are more viable in terms of cost and complexity.

The combination of Graph-RAG and RL, or even just one of these techniques, is more relevant for smaller models, which benefit from instructions with context and more detailed guidance, while larger models tend to perform better with more concise data or, in

some cases, no additional data at all. Even with the application of techniques like Graph-RAG, larger models maintain high effectiveness in English but exhibit performance drops when applied to multilingual datasets.

## 5. Related Works

Recent studies combining LLMs with KGs have focused on models like OpenAI's GPT-3.5 and Meta's Llama. GPT-3.5 has been applied in areas such as engineering education [Yang et al. 2023], text classification [Shi et al. 2023], and node classification in graph structures [Li et al. 2024]. GPT-4 has been used in recommendation systems and biomedical studies [Xu et al. 2024, Guan et al. 2023]. Meta's Llama2 models have shown effectiveness in processing complex graphs, with applications in vision systems, academic databases, and digital news domains [Gouidis et al. 2024, Hu et al. 2024, Wu et al. 2024]. Chain of Thought (CoT) prompting and GNN techniques have also been integrated with LLMs for improving model interpretability and processing structured knowledge from KGs [Guan et al. 2023, Xu et al. 2023]. Techniques like PCA, UMAP, and prompt methods further integrate LLMs into the visual and structural domains of KGs, enhancing zero-shot learning [Gouidis et al. 2024, Alfasi et al. 2024]. In RL, approaches like RLHF and RLAIF have demonstrated improvements in summarization, negotiation dialogues, and domain knowledge applications [Roit et al. 2023, Kwon et al. 2024, Mandi et al. 2023]. Although effective, RLHF and finetuning are expensive and nearly unfeasible for most experiments due to the significant computational and financial resources required [Ouyang et al. 2022, Nguyen et al. 2023]. Persistent issues like biases, toxicity, and hallucinations remain critical in both KGs and RL contexts [Gouidis et al. 2024, Xu et al. 2024, McKenna et al. 2023]. Differently from previous works, our study addresses scalability high costs associated with the use of very large model and traditional techniques fine-tuning, by combineing RAG with RL. We demonstrate the effectiveness of this approach for multi-label classification and information extraction using domain-specific KGs and datasets.

## 6. Conclusions and Future Work

This study demonstrates the feasibility and effectiveness of integrating LLMs with Graph-RAG to enhance multi-label classification and information extraction. Experiments conducted with variations of the GPT and Llama models, combined with the use of KGs and an RL agent, revealed significant improvements in the performance of smaller models, such as Llama3 8b, especially when combined with Graph-RAG. The combination of smaller LLMs and Graph-RAG reduces the occurrence of "hallucinations", contributing to superior accuracy and effectiveness, even in multilingual contexts. These outcomes suggest a promising future for not so large LLM's, especially in organizations facing data privacy constraints and computational resource limitations. As future research directions, we envision the exploitation of more diverse KGs and the investigation of RL techniques to further improve results of complex tasks. Furthermore, additional studies could apply our proposal to low resource languages, for expanding its accessibility and applicability.

**Acknowledgements:** This work was supported by a 2022 CNPq Universal grant, FAPESC grant 2021TR1510, the Print CAPES-UFSC Automation 4.0 Project, and indirectly by the Céos project, financed by the Public Ministry of Santa Catarina State (MPSC).

## References

[Alfasi et al. 2024] Alfasi, D., Shapira, T., and Barr, A. B. (2024). Unveiling hidden links between unseen security entities. *arXiv preprint arXiv:2403.02014*.

[Alizadeh et al. 2023] Alizadeh, K., Mirzadeh, I., Belenko, D., Khatamifard, K., Cho, M., Del Mundo, C. C., Rastegari, M., and Farajtabar, M. (2023). Llm in a flash: Efficient large language model inference with limited memory. *arXiv preprint arXiv:2312.11514*.

[Beckhauser and Fileto 2024] Beckhauser, W. and Fileto, R. (2024). Can a simple customer review outperform a feature set for predicting churn? In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 117–128, Porto Alegre, RS, Brasil. SBC.

[Bruno et al. 2023] Bruno, A., Mazzeo, P. L., Chetouani, A., Tliba, M., and Kerkouri, M. A. (2023). Insights into classifying and mitigating llms' hallucinations. *arXiv arXiv:2311.08117*.

[Chen et al. 2024] Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. (2024). Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *ArXiv*, abs/2402.03216.

[Erickson 2018] Erickson, A. (2018). Comparative analysis of the eu's gdpr and brazil's lgpd: Enforcement challenges with the lgpd. *Brook. J. Int'l L.*, 44:859.

[Gao et al. 2023] Gao, P., Han, J., Zhang, R., Lin, Z., Geng, S., Zhou, A., Zhang, W., Lu, P., He, C., Yue, X., et al. (2023). Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*.

[Gouidis et al. 2024] Gouidis, F., Papantoniou, K., Patkos, K. P. T., Argyros, A., and Plexousakis, D. (2024). Fusing domain-specific content from large language models into knowledge graphs for enhanced zero shot object state classification. *arXiv arXiv:2403.12151*.

[Guan et al. 2023] Guan, Z., Wu, Z., Liu, Z., Wu, D., Ren, H., Li, Q., Li, X., and Liu, N. (2023). Cohortgpt: An enhanced gpt for participant recruitment in clinical study. *arXiv preprint arXiv:2307.11346*.

[Hartmann et al. 2023] Hartmann, J., Heitmann, M., Siebert, C., and Schamp, C. (2023). More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87.

[He et al. 2023] He, X., Bresson, X., Laurent, T., Perold, A., LeCun, Y., and Hooi, B. (2023). Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. In *ICLR*.

[Hu et al. 2024] Hu, S., Zou, G., Yang, S., Zhang, B., and Chen, Y. (2024). Large language model meets graph neural network in knowledge distillation. *arXiv preprint arXiv:2402.05894*.

[Krugmann and Hartmann 2024] Krugmann, J. O. and Hartmann, J. (2024). Sentiment analysis in the age of generative ai. *Customer Needs and Solutions*, 11(1):1–19.

[Kwon et al. 2024] Kwon, D., Weiss, E., Kulshrestha, T., Chawla, K., Lucas, G. M., and Gratch, J. (2024). Are llms effective negotiators? systematic evaluation of the multifaceted capabilities of llms in negotiation dialogues. *arXiv preprint arXiv:2402.13550*.

[Li et al. 2024] Li, R., Li, J., Han, J., and Wang, G. (2024). Similarity-based neighbor selection for graph llms. *arXiv preprint arXiv:2402.03720*.

[Liu et al. 2024] Liu, X., Li, P., Huang, H., Li, Z., Cui, X., Liang, J., Qin, L., Deng, W., and He, Z. (2024). Fakenewsgpt4: Advancing multimodal fake news detection through knowledge-augmented lvlms. *arXiv preprint arXiv:2403.01988*.

[Liu et al. 2023] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. (2023). Gpt understands, too. *AI Open*.

[Mandi et al. 2023] Mandi, Z., Jain, S., and Song, S. (2023). Roco: Dialectic multi-robot collaboration with large language models. *arXiv preprint arXiv:2307.04738*.

[McKenna et al. 2023] McKenna, N., Li, T., Cheng, L., Hosseini, M. J., Johnson, M., and Steedman, M. (2023). Sources of hallucination by large language models on inference tasks. *arXiv preprint arXiv:2305.14552*.

[Nguyen et al. 2023] Nguyen, H. A., Stec, H., Hou, X., Di, S., and McLaren, B. M. (2023). Evaluating chatgpt's decimal skills and feedback generation in a digital learning game. In *European Conference on Technology Enhanced Learning*, pages 278–293. Springer.

[Ouyang et al. 2022] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

[Pan et al. 2023] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. (2023). Unifying large language models and knowledge graphs: A roadmap. *ArXiv*, abs/2306.08302.

[Roit et al. 2023] Roit, P., Ferret, J., Shani, L., Aharoni, R., Cideron, G., Dadashi, R., Geist, M., Girgin, S., Hussenot, L., Keller, O., et al. (2023). Factually consistent summarization via reinforcement learning with textual entailment feedback. *arXiv preprint arXiv:2306.00186*.

[Shen et al. 2024] Shen, W., Li, C., Chen, H., Yan, M., Quan, X., Chen, H., Zhang, J., and Huang, F. (2024). Small llms are weak tool learners: A multi-llm agent. *arXiv preprint arXiv:2401.07324*.

[Shi et al. 2023] Shi, Y., Ma, H., Zhong, W., Tan, Q., Mai, G., Li, X., Liu, T., and Huang, J. (2023). Chatgraph: Interpretable text classification by converting chatgpt knowledge to graphs. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 515–520. IEEE.

[Shridhar et al. 2023] Shridhar, K., Sinha, K., Cohen, A., Wang, T., Yu, P., Pasunuru, R., Sachan, M., Weston, J., and Celikyilmaz, A. (2023). The art of llm refinement: Ask, refine, and trust. *arXiv preprint arXiv:2311.07961*.

[Team et al. 2023] Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., et al. (2023). Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

[Wang et al. 2023] Wang, Q., Gao, Z., and Xu, R. (2023). Graph agent: Explicit reasoning agent for graphs. *arXiv preprint arXiv:2310.16421*.

[Wu et al. 2024] Wu, H., Zhang, Y., Han, Z., Hou, Y., Wang, L., Liu, S., Gong, Q., and Ge, Y. (2024). Quartet logic: A four-step reasoning (qlfr) framework for advancing short text classification. *arXiv preprint arXiv:2401.03158*.

[Xu et al. 2024] Xu, D., Zhang, Z., Lin, Z., Wu, X., Zhu, Z., Xu, T., Zhao, X., Zheng, Y., and Chen, E. (2024). Multi-perspective improvement of knowledge graph completion with large language models. *arXiv preprint arXiv:2403.01972*.

[Xu et al. 2023] Xu, H., Gao, Y., Hui, Z., Li, J., and Gao, X. (2023). Language knowledge-assisted representation learning for skeleton-based action recognition. *arXiv preprint arXiv:2305.12398*.

[Xue et al. 2024] Xue, F., Fu, Y., Zhou, W., Zheng, Z., and You, Y. (2024). To repeat or not to repeat: Insights from scaling llm under token-crisis. *Advances in Neural Information Processing Systems*, 36.

[Yang et al. 2023] Yang, Y., Chen, S., Zhu, Y., Zhu, H., and Chen, Z. (2023). Knowledge graph empowerment from knowledge learning to graduation requirements achievement. *Plos one*, 18(10):e0292903.

[Yao et al. 2024] Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., and Zhang, Y. (2024). A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.