Flowchart-Based Decision Making with Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) are widely used for conversational systems, but they face significant challenges in interpretability of dialogue flow and reproducibility of expert knowledge. To address this, we propose a novel method that extracts flowcharts from dialogue data and incorporates them into LLMs. This approach not only makes the decision-making process more interpretable through visual representation, but also ensures the reproducibility of expert knowledge by explicitly modeling structured reasoning flows. By evaluating on dialogue datasets, we demonstrate that our method effectively reconstructs expert decision-making paths with high precision and recall scores. These findings underscore the potential of flowchart-based decision making to bridge the gap between flexibility and structured reasoning, making chatbot systems more interpretable for developers and end-users.

1 Introduction

The rapid advancement of large language models (LLMs) has enabled their widespread adoption in real-world applications, including customer support, medical diagnosis, and security incident response (Sun et al., 2024; Hu et al., 2024; Wang et al., 2024; Liu, 2024; Hays and White, 2024; Ouyang et al., 2022). In particular, conversational agent systems use LLMs for more flexible and adaptive interactions compared to traditional rule-based approaches(Xi et al., 2025; Adamopoulou and Moussiades, 2020; Hussain et al., 2019). However, two fundamental challenges remain: *interpretability of dialogue flows* and *reproducibility of expert knowledge*.

First, LLM-driven dialogues lack interpretability, making it difficult for system administrators to control and manage interactions. While retrieval-augmented generation (RAG) and fine-tuning can improve LLM knowledge, the decision-making process remains opaque (Lewis et al., 2020; Gao et al., 2023; Barnett et al., 2024; Ding et al., 2023; Rafailov et al., 2024; Ovadia et al., 2023).

Second, the reproducibility of expert knowledge is critical in high-stakes domains such as medical diagnosis and troubleshooting. Experts rely on structured decision-making processes, but existing LLM-based conversational agent systems lack explicit modelling of these reasoning flows, leading to inconsistencies. Moreover, expert knowledge is constantly evolving, making it challenging to keep dialogue systems up-to-date with the latest domain expertise (Hudeček and Dušek, 2023; Sekulić et al., 2024; Ulmer et al., 2024).

To address these challenges, we propose a novel method that extracts flowcharts from dialogue data and incorporates them into LLMs. This approach not only makes the decision-making process more interpretable through visual representation, but also ensures the reproducibility of expert knowledge by explicitly modeling structured reasoning flows. The flowchart-based representation allows system administrators to easily monitor, update, and maintain the dialogue system while preserving the expert decision-making patterns embedded in the training data.

Our method consists of three main steps: extracting key decision points from dialogues by identifying expert questions and suggestions, structuring them into logical decision flows, and constructing a refined flowchart by aggregating these flows.

Contributions. The key contributions of this paper are:

- We propose a novel method for extracting and visualizing expert decision flows from dialogue data using flowcharts.
- We evaluate the reproducibility of expert decision flows using the Flo-dial dataset, measuring precision, recall, and F1-score.

2 Proposed Method

Flowcharts are effective tools for visualizing complex processes (Gilbreth and Gilbreth, 2023). Our method extract decision-making flows as flowcharts from dialogue data between expert operators and users, as illustrated in Figure 1. This approach is particularly effective in high-stakes applications such as disease diagnosis and troubleshooting, where operators identify and resolve users' issues. Our method consists of the following three steps: (1) Question and Suggestion Extraction, (2) Dialogue Flow Extraction, and (3) Flowchart Construction.

Step 1: Question and Suggestion Extraction

The first step involves identifying key decision points within dialogues, specifically questions posed by experts and suggestions provided in response to user inputs.

Each dialogue is processed using an LLM to categorize utterances into two types: (1) Questions: Information-seeking utterances that guide the user (e.g., "Do you have a fever?"). (2) Suggestions: Recommendations, diagnoses, or conclusions provided by the expert (e.g., "You might have a cold."). The classification process also filters out irrelevant statements such as greetings and acknowledgments.

Since this classification is conducted independently for each dialogue, the extracted lists of questions and suggestions often contain significant semantic redundancy. To address this, similar questions (e.g., "Are you experiencing dizziness?" and "Do you feel lightheaded?") are grouped using an LLM-based merging process. Suggestions with equivalent meanings but different expressions are also normalized to ensure consistency in the flowchart representation. This allows us to extract questions and suggestions from all dialogues based on a unified perspective.

The prompts used for extracting and merging questions, as well as their equivalents for suggestions, are summarized in Table 1.

Step 2: Dialogue Flow Extraction

After identifying questions and suggestions, the next step is to structure their connections within dialogues. Using the extracted elements from Step 1, unstructured dialogues are converted into ordered sequences of questions, user responses, and operator suggestions.

First, each question posed by the operator in a dialogue is mapped to the corresponding question ID from the extracted list. Next, for each identified question, the user's response is extracted and normalized into short categorical answers such as *"Yes"*, *"No"*, or *"I don't know"*. This normalization ensures that decision branches in the flowchart are clearly defined based on user input. Finally, we identify the operator's suggestion following each sequence of questions and responses, using the extracted list of suggestions. By structuring dialogues in this manner, we convert conversational interactions into an ordered series of questions, user responses, and expert suggestions, creating a structured representation suitable for flowchart construction.

The prompts used for mapping questions, user responses, and suggestions are summarized in Table 2.

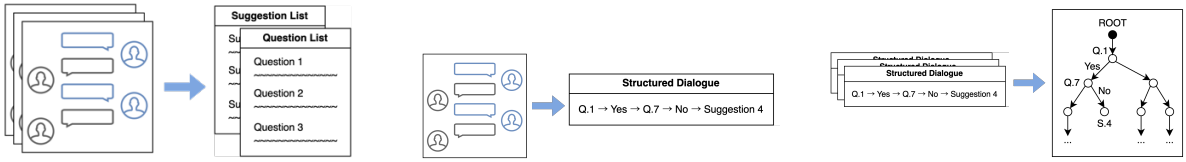
Step 3: Flowchart Construction

In this step, the structured sequences of questions, user responses, and expert suggestions from Step 2 are aggregated to form the final flowchart. The goal is to create a decision tree that accurately represents expert decision-making while removing infrequent transitions for clarity.

The flowchart is represented as a directed graph, where nodes correspond to extracted questions or suggestions, and edges represent transitions based on user responses. The construction process begins by sorting questions based on their frequency as parent nodes, ensuring that frequently asked questions are prioritized. The most common question is linked to a root node ("ROOT"), establishing the starting point of the flowchart.

For each parent node, child nodes are determined using recorded transitions from the dialogue data. These transitions are sorted by frequency, and low-frequency transitions are discarded to reduce noise. User response types (e.g., *"Yes"*, *"No"*, or *"I don't know"*) are explicitly recorded as edge attributes to clarify decision paths. If a child node represents a suggestion rather than a follow-up question, it is registered as a terminal node, ensuring a clear distinction between inquiry and decision points.

Throughout this process, cyclic dependencies are checked and removed to maintain a well-structured hierarchy. Additionally, if multiple edges exist between the same nodes with different response labels, they are merged into a single edge while preserving all response variations. This ensures that the resulting flowchart remains compact and interpretable while effectively reconstructing expert reasoning patterns. The complete flowchart construction process is outlined in Algorithm 1.



(a) Question and suggestion extraction (b) Dialogue flow extraction (c) Flowchart construction

Figure 1: Overview of the proposed method: Our approach (a) first extracts questions and suggestions from dialogue data, (b) then structures these elements into sequential flows, and (c) finally aggregates them into a comprehensive flowchart, enabling systematic transformation of expert dialogues into structured decision flowchart.

Table 1: Prompts used for question extraction. Similar templates are used for suggestion extraction as well.

Task	Prompt Template
Question Extraction	`Refer to the following conversation between USER and OPERATOR to identify questions that the operator should ask the user, together with sample questions. The questions should be as closed-ended as possible, eliciting information useful for assisting the user. Split multi-topic questions into separate simple ones.` `## conversation {text}`

Question Merging	`Merge the two provided sets of questions into a single, consolidated set.`
	
	`- Include all unique questions from both sets.`
	`- Merge semantically similar questions into a single entry.`
	`- Choose the simpler and clearer version for merged questions.`
	`## questions 1 {questions1}`
	`## questions 2 {questions2}`

3 Experiments

We conducted experiments using GPT-4o-2024-08-06 as the LLM to evaluate the effectiveness of our proposed method (Achiam et al., 2023).

3.1 Dataset

The FloDial dataset consists of 2,738 troubleshooting dialogues between users and customer support operators, collected via Amazon Mechanical Turk (Raghu et al., 2021). These dialogues are based on 12 distinct troubleshooting flowcharts covering technical issues with cars and laptops. The dataset is released under the CDLA-Sharing-1.0 license.

On average, each flowchart corresponds to 228.2 dialogues and contains 15.25 unique decision paths, representing different troubleshooting scenarios. Dialogues have an average of 7.19 turns, where users describe initial fault symptoms, and operators diagnose issues through targeted questions.

Our experiments focused on reconstructing flowcharts from these dialogues.

3.2 Evaluation Metrics

To evaluate our method, we compare the decision-making paths in the generated and ground truth FloDial flowcharts, measuring their overlap to assess structural similarity.

We extract all possible paths from both the ground truth and generated flowcharts, where each path represents a sequence of questions, responses, and suggestions. These paths are then compared using an LLM-based similarity assessment to determine whether they represent the same problem-solving flow (Zheng et al., 2023; Gu et al., 2024).

To quantify the accuracy of the generated flowcharts, we compute precision ($P = M/G$), recall ($R = M/T$), and F1 score ($F_1 = 2PR/(P + R)$), where M is the number of matched paths, G is the total number of generated paths, and T is the total number of ground truth paths. Matched paths are those generated paths that were successfully matched to a ground truth path.

3.3 Results

The evaluation results of our flowchart reconstruction method are summarized in Table 3. We assessed the precision, recall, and F1-score for each of the 12 flowcharts in the FloDial dataset.

Overall, our method demonstrated a strong ability to capture the complete decision-making paths present in the ground truth flowcharts, as evidenced by the high recall scores. In several cases, such as "Engine Overheats", the recall reached a perfect 1.000, indicating that all relevant decision paths were successfully captured. This highlights our

Table 2: Prompts used for dialogue flow extraction.

Task	Prompt Template

Question Mapping	`Please select the IDs of the question contained in the following utterance from the list of questions. A single utterance can contain multiple questions with different meanings, or none at all. If no questions are included, respond with an empty list.`
	`## Utterance {utterance}`
	`## Questions {question_list}`

User Response Mapping	`Please extract the user's response to the following question below based on the user's most recent utterances in the following conversation. The answer should be as concise as possible (Yes, No, I don't know, etc.).`
	`## Conversation {conversation}`
	`## Question {question}`

Suggestion Mapping	`Refer to the following conversation and select the solution IDs proposed by the OPERATOR to the USER from the list of solutions.`
	`## Conversation {conversation}`
	`## Solutions {solution_list}`

Algorithm 1 Flowchart Construction

Require: Sequences (Q, A, C) where
Q: question,
A: user response,
C: next question or suggestion
Ensure: Directed Graph G representing the flowchart
$G \leftarrow$ initialize graph
$cnt \leftarrow$ count occurrences of (Q, A, C)
$pcnt \leftarrow$ count occurrences of each Q as parent
$plist \leftarrow$ sort $pcnt$ by frequency (desc)
add_edge(G, "ROOT", plist[0], "START")
for each p in $plist$ **do**
  $ccnt \leftarrow$ filter cnt where key $= (p, *, *)$
  $clist \leftarrow$ sort $ccnt$ by frequency (desc)
  for each (p, r, c) in $clist$ **do**
    if $c \notin G$ or no cycle(G, c, p) **then**
      add_node(G, c)
      if edge(p, c) exists **then**
        update_edge(G, p, c, r)
      else
        add_edge(G, p, c, r)
      end if
    end if
  end for
end for
return G

Table 3: Evaluation results of flowchart reconstruction.

Flowchart	Precision	Recall	F1-score
Brake Problem	0.750	0.947	0.837
Car Electrical Failure	0.722	0.929	0.813
Car Steering	0.900	0.947	0.923
Car Won't Start	0.708	1.000	0.829
Engine Overheats	0.809	1.000	0.894
Laptop Battery	0.666	1.000	0.799
Laptop Drive	0.800	0.800	0.800
Laptop Overheating	0.846	0.846	0.846
LCD Problem	0.714	1.000	0.833
Power	0.722	0.867	0.788
Ticking	0.900	0.750	0.818
Wireless	0.666	0.933	0.777
Average	0.767	0.918	0.830

method's effectiveness in ensuring comprehensive coverage of expert reasoning.

However, the precision scores were relatively lower, suggesting that the generated flowcharts included some extraneous paths. This is likely due to challenges in merging similar questions and suggestions, leading to redundant nodes. For example, the "Laptop Battery" flowchart exhibited a precision of 0.666, indicating room for improvement in distinguishing between similar decision paths.

The F1-scores, consistently above 0.80 across various flowcharts, demonstrate the effectiveness of the proposed method in reconstructing expert decision-making processes. While there is room for improving precision by reducing redundant paths, the high recall scores validate our approach's ability to capture comprehensive troubleshooting knowledge. This structured representation of dialogue flows enables the development of more explainable and reliable chatbot systems.

4 Conclusion

We introduced a method for extracting and visualizing expert decision flows from dialogue data using flowcharts, addressing interpretability and reproducibility challenges in conversational agent systems. Utilizing the FloDial dataset, our approach effectively captures comprehensive decision paths, as evidenced by high recall scores.

These findings underscore the potential of flowchart-based decision making to enhance the transparency and reliability of chatbot systems, making them more interpretable for developers and end-users.

5 Limitations

While our method for extracting and visualizing expert decision flows using flowcharts offers significant advancements in interpretability and reproducibility, several limitations must be acknowledged.

Firstly, our approach does not fully account for scenarios where questions can be asked in any order, such as asking for a name and gender. Although Step 3 prioritizes statistically frequent patterns, it does not explicitly incorporate the concept of "order independence," which may lead to inefficiencies. Introducing a flow alignment step between Steps 2 and 3 could address this limitation.

Secondly, the method focuses on extracting structured decision-making processes, which may not fully capture the nuances of complex dialogues. This could limit its effectiveness in domains where context dependency is high, potentially restricting its applicability in such areas.

Thirdly, our experiments were conducted exclusively with GPT-4, which may not generalize to other models. The task of merging questions is particularly challenging for LLMs, and achieving high accuracy may require models with capabilities similar to GPT-4.

Additionally, our evaluation was limited to the FloDial dataset, as it is the only dataset with corresponding ground truth flowcharts. This dependency on the quality and comprehensiveness of dialogue data means that incomplete datasets may result in flowcharts that do not fully reflect expert reasoning processes.

Finally, we have not yet evaluated the performance of chatbots incorporating these flowcharts. While we assume that accurate flowcharts facilitate the creation of guided chatbots, future work should include performance metrics such as Winrate to validate this assumption across other datasets like MultiWOZ (Budzianowski et al., 2018).

Despite these limitations, our method provides a valuable framework for enhancing the transparency and reliability of chatbot systems, offering a structured approach to decision-making that can be further refined and expanded upon in future research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Eleni Adamopoulou and Lefteris Moussiades. 2020. Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006.

Scott Barnett, Stefanus Kurniawan, Srikanth Thudumu, Zach Brannelly, and Mohamed Abdelrazek. 2024. Seven failure points when engineering a retrieval augmented generation system. *Preprint*, arXiv:2401.05856.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Frank B. Gilbreth and L. M. Gilbreth. 2023. Process charts: First steps in finding the one best way to do work. *Transactions of the American Society of Mechanical Engineers*, 43:1029–1043.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Sam Hays and Jules White. 2024. Employing llms for incident response planning and review. *Preprint*, arXiv:2403.01271.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. 2024. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Vojtěch Hudeček and Ondřej Dušek. 2023. Are llms all you need for task-oriented dialogue? *arXiv preprint arXiv:2304.06556*.

Shafquat Hussain, Omid Ameri Sianaki, and Nedal Ababneh. 2019. A survey on conversational agents/chatbots classification and design techniques. In *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*, pages 946–956. Springer.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Zefang Liu. 2024. Multi-agent collaboration in incident response with large language models. *Preprint*, arXiv:2412.00652.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Dinesh Raghu, Shantanu Agarwal, Sachindra Joshi, and Mausam. 2021. End-to-end learning of flowchart grounded task-oriented dialogs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ivan Sekulić, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, André Ferreira Manso, and Roland Mathis. 2024. Reliable llm-based user simulator for task-oriented dialogue systems. *arXiv preprint arXiv:2402.13374*.

Yiyou Sun, Junjie Hu, Wei Cheng, and Haifeng Chen. 2024. Chatbot meets pipeline: Augment large language model with definite finite automaton. *arXiv preprint arXiv:2402.04411*.

Dennis Ulmer, Elman Mansimov, Kaixiang Lin, Justin Sun, Xibin Gao, and Yi Zhang. 2024. Bootstrapping llm-based task-oriented dialogue agents via self-talk. *arXiv preprint arXiv:2401.05033*.

Sheng Wang, Zihao Zhao, Xi Ouyang, Tianming Liu, Qian Wang, and Dinggang Shen. 2024. Interactive computer-aided diagnosis on medical image using large language models. *Communications Engineering*, 3(1):133.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Prompts used for evaluation

Table 4: Prompt used for LLM-based path similarity evaluation.

Prompt Template

`Determine if two paths represent the same problem-solving flow. If similar questions and answers are exchanged in the same order and the final solution is similar, regard them as equivalent.`
`Compare the following two troubleshooting flows and determine if they represent the same problem-solving process.`
`## Path 1: {path1}`
`## Path 2: {path2}`

B Example of a generated flowchart

To demonstrate the effectiveness of our method, Figure 2 presents an example flowchart generated for car steering troubleshooting, which achieved the highest F1 score in our evaluation experiments. As a comparison, Figure 3 shows the ground truth flowchart for car steering troubleshooting.

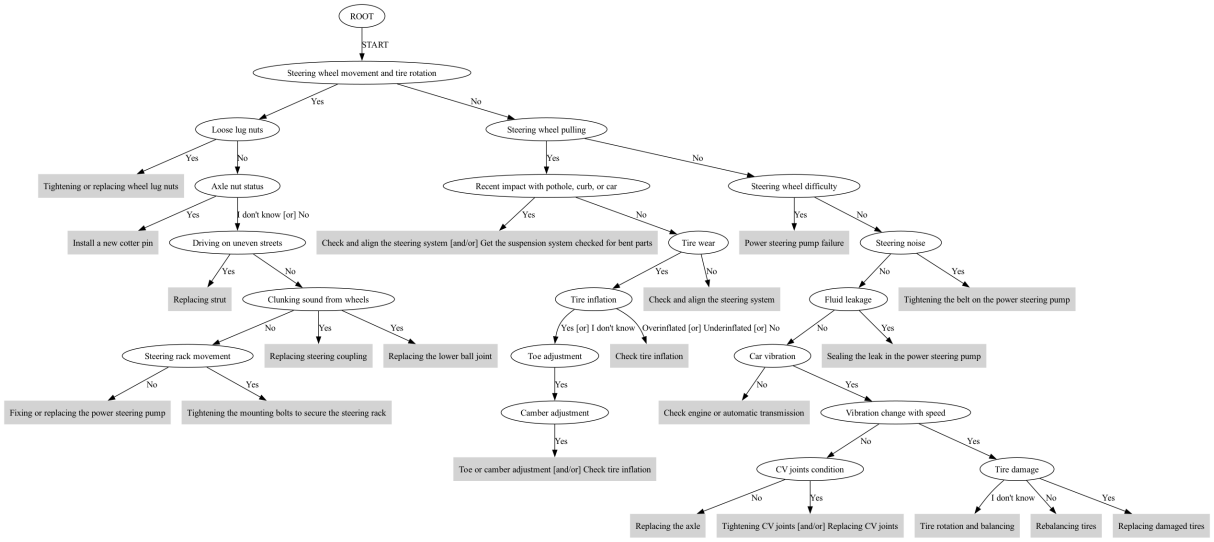


Figure 2: Example of a generated flowchart for car steering troubleshooting. The flowchart captures the main decision points and solutions.

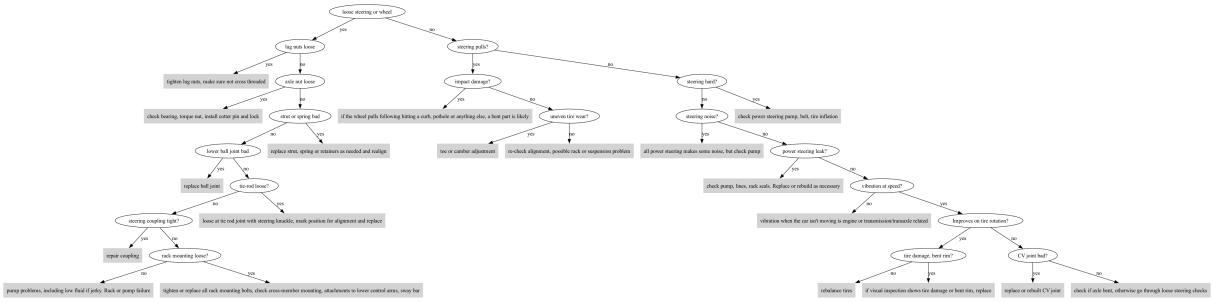


Figure 3: Ground truth flowchart for car steering troubleshooting.