Exploring Large Language Models for Procedural Extraction from Documents

Anisa Rula[1,*], Jennifer D'Souza[2]

[1] *University of Brescia, Brescia, Italy*

[1] *TIB Leibniz Information Centre for Science and Technology, Hannover, Germany*

Abstract

Recent advancements in Natural Language Processing (NLP), notably the emergence of extensive language models pre-trained on vast datasets, are opening new avenues in Knowledge Engineering. This study delves into the utilization of these large language models (LLMs) in two learning scenarios - zero-shot and in-context learning - to address the extraction of procedures from unstructured PDF texts through incremental question-answering techniques. Specifically, we employ the cutting-edge GPT-4 (Generative Pre-trained Transformer 4) model, alongside two variations of in-context learning methodologies. These methods incorporate an ontology with definitions of procedures and steps, as well as a limited set of samples for few-shot learning. Our investigation underscores the potential of this approach and underscores the significance of tailored in-context learning adaptations. These adjustments hold promise in mitigating the challenge of acquiring adequate training data, a common obstacle in deep learning-based NLP methods for procedure extraction.

Keywords

Procedural knowledge, knowledge graphs, large language models, knowledge capture

1. Introduction

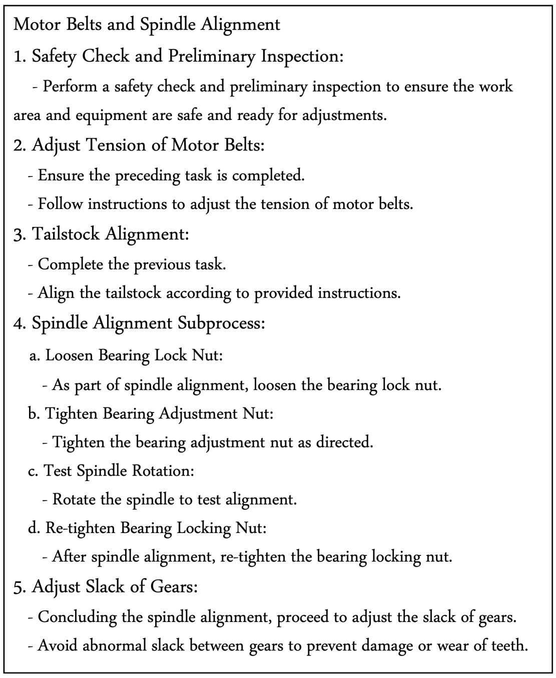
Extracting complex knowledge from unstructured sources is a challenge, particularly focusing on industrial troubleshooting documents. These documents often contain detailed procedures represented as sequences of steps, which vary in textual form, making it challenging for automated algorithms to identify and organize the relevant information accurately. Despite advancements in Natural Language Processing (NLP), the scarcity of training data remains a significant obstacle for machine learning approaches. Consequently, novel methods are emerging, such as interactive dialogues and language models, to address this challenge [1].

The paper emphasizes the importance of extracting relevant procedures, using the example of a shop floor operator needing to follow maintenance procedures for gear head lathe machinery. It outlines a typical sequence of activities involved in maintenance and highlights the importance of correct execution for optimal machine performance (see Figure 1). The shop floor worker may need to answer some questions for which a simple keyword-based search in the document is not sufficient:

SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy

[*] Corresponding author.

anisa.rula@unibs.it (A. Rula); jennifer.dsouza@tib.eu (J. D'Souza)



Motor Belts and Spindle Alignment

1. Safety Check and Preliminary Inspection:
   - Perform a safety check and preliminary inspection to ensure the work area and equipment are safe and ready for adjustments.
2. Adjust Tension of Motor Belts:
   - Ensure the preceding task is completed.
   - Follow instructions to adjust the tension of motor belts.
3. Tailstock Alignment:
   - Complete the previous task.
   - Align the tailstock according to provided instructions.
4. Spindle Alignment Subprocess:
   a. Loosen Bearing Lock Nut:
      - As part of spindle alignment, loosen the bearing lock nut.
   b. Tighten Bearing Adjustment Nut:
      - Tighten the bearing adjustment nut as directed.
   c. Test Spindle Rotation:
      - Rotate the spindle to test alignment.
   d. Re-tighten Bearing Locking Nut:
      - After spindle alignment, re-tighten the bearing locking nut.
5. Adjust Slack of Gears:
   - Concluding the spindle alignment, proceed to adjust the slack of gears.
   - Avoid abnormal slack between gears to prevent damage or wear of teeth.

Figure 1: Example of a procedure.

- *What are the steps involved in performing routine maintenance on machinery gear head lathe?*
- *Are there any sub-procedures or specialised steps for troubleshooting specific issues during maintenance procedures for machinery gear head lathe?*

To address the limitations of keyword-based search methods, it's essential to extract and depict procedural knowledge using a vocabulary that encompasses domain-specific terms and concepts. This extraction and representation process is supported by an in-context learning strategy, allowing for the customization of large language models (LLMs) with minimal training data. Integrating this approach makes knowledge engineering more accessible to individuals lacking expertise in formal representation languages. This structured knowledge can then be queried for semantic procedural data, enabling responses to queries that were previously impossible with unstructured text. By querying over semantically structured procedures, computers can intelligently assist users in efficiently managing, comprehending, and executing procedures.

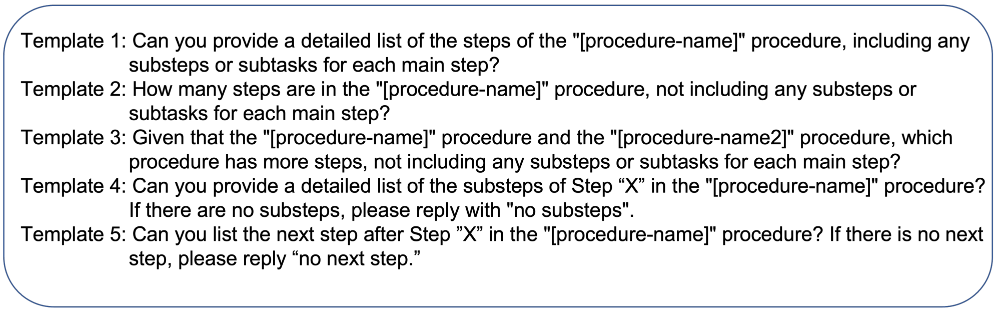
The paper is structured as follows: section 2, we propose our approach and discuss our experimental results in section 3. Finally, a brief discussion on related work is offered in section 4 and concluding remarks in section 5.

2. Approach

This section provides an overview of the methodology adopted to construct different versions of question-answering systems through in-context learning. The primary focus during the design of conversational interactions was on extracting procedures, including their constituent steps and substeps as these form the fundamental components of any procedure. To create the conversational system, we employed GPT-4 (Generative Pre-trained Transformer 4.0) hosted on ChatGPT-Plus, along with the https://askyourpdf.com/ plugin for GPT-4. This choice was based on GPT-4 as a cutting-edge standard in LLMS. However, given that raw pre-trained language models may not seamlessly align with specific tasks, we employed in-context learning through prompting to customise the native model to varying extents. The following sections will provide detailed insights into the strategies adopted for formulating query templates and the customised models functioning as conversational systems.

2.1. Prompting and questioning

Figure 2 illustrates five distinct queries formulated for information extraction purposes. The questions, and therefore construction, are posed incrementally. First, we ask questions about the list of steps and substeps and then we ask questions regarding aggregations and comparisons, and finally, the precedence relations among steps. We discuss shortly each of these templates:



Template 1: Can you provide a detailed list of the steps of the "[procedure-name]" procedure, including any substeps or subtasks for each main step?
Template 2: How many steps are in the "[procedure-name]" procedure, not including any substeps or subtasks for each main step?
Template 3: Given that the "[procedure-name]" procedure and the "[procedure-name2]" procedure, which procedure has more steps, not including any substeps or subtasks for each main step?
Template 4: Can you provide a detailed list of the substeps of Step “X” in the "[procedure-name]" procedure? If there are no substeps, please reply with "no substeps".
Template 5: Can you list the next step after Step ”X” in the "[procedure-name]" procedure? If there is no next step, please reply “no next step.”

Figure 2: Question templates and their ordering

- *Template 1 (List):* Requests a detailed list of steps for a specific procedure, including any step or substep for each main step.
- *Template 2 (Counting):* Asks for the total number of steps within a procedure.
- *Template 3 (Comparison):* Aims to identify the procedure with the maximum number of steps among a set of procedures.
- *Template 4 (Nested Procedures):* Asks about the presence of sub-procedures or substeps within the main procedure.
- *Template 5 (Sequence):* Asks for the next step after a specific step in a procedure and provides the step that comes immediately after the specified step.

These questions were structured incrementally to address the model’s limitations in handling complex queries that encompass the entire procedural structure. This incremental approach shares similarities with the iterative process of crafting conceptual models often involving interactions with domain experts. It paves the way for versatile pipelines by combining diverse incremental inquiries.

2.2. In-context learning customisations

"Contextual learning" involves training models to understand the contextual environment within which information is presented. This context can encompass surrounding text, images, or other data facets. Contextual learning is particularly important for tasks like NLP, where the meaning of a word or phrase can vary based on its surrounding context.

Learning approaches: Raw vs Zero-shot vs 2-shot. The initial learning approach aligns with "zero-shot learning." where the pre-trained GPT-4 model can generalise its knowledge to tasks it has not been explicitly trained on but gains its understanding by providing the model with some initial context. In contrast "2-shot learning" falls between traditional supervised learning and zero-shot learning. This approach involves training a machine learning model with just two labelled examples per class, enabling the model to generalise and make predictions for new instances even with limited labelled data.

Contextual knowledge definitions. Contextual knowledge is provided through the identification of the specific domain and intensional definitions of the procedure elements to be extracted. Relying on intensional definitions offers the advantage of compactness without requiring the provision of examples. These definitions consist of the main concepts and properties defined in the ontology along with definitions of Procedure, Steps and Substeps. Each definition is labelled with the question it was used for. These choices were made to minimise external knowledge while providing an initial empirical assessment of using intentional definitions in the customisation of pre-trained models.

3. Evaluations

In this section, we discuss the results obtained by leveraging ChatGPT4 for procedural text mining w.r.t. the 5 prompting scenarios introduced in subsection 2.1.

3.1. Datasets

To understand the effectiveness of our approach we select four domains where public data are easily accessible and which cover all the challenges.

- Photography, manuals that provide instructions for operating, maintaining, and troubleshooting cameras, covering settings, capturing images or videos, lens care, and handling various scenarios.
- Manufacturing, manuals that provide instructions for production, quality control, assembly, maintenance, and safety in various manufacturing operations.
- Medicine, manuals that provide instructions for dental instrument usage, sterilisation, X-rays, oral hygiene instructions, and emergency protocols.
- Agriculture, manuals that provide instructions for the operation, maintenance, and safety protocols of farm equipment.

We now give details on how we extracted the documents from each data source. First, we examine all the procedures defined in the manuals. Second, we specifically choose procedures that adhere to a structured format suitable for enumeration, such as numbering, bullet points, or clear indentation. Third, we prioritise procedures that are only on one page or at most spanned across two consecutive pages. For each domain, we extract three examples of procedures either from the same manual or different manuals.

3.2. Qualitative Evaluations

The qualitative evaluations are discussed in terms of 9 observations presented as questions.

1. *With the 2-shot in-context learning setting, can one tailor the model to respond in a certain way or a certain format?* E.g., for prompt scenario 5 in medicine, when asked for the next instruction in sequence, we would like the agent to reply with just the name or the sentence corresponding to the next instruction. However, the model in the "raw" setting seems to respond with the next instruction, but also with the substeps or additional information like notes or warnings related to the instruction. See response where it also adds the sentence "Refer to page 17 of the document." which is additional information in the context of the instruction sentence "CONTROL BOX INSTALLATION". However, in the 2-shot setting, the model after seeing reference examples, responds with just the essential information for the same query as shown in context. See response now reads "... the next step after Step 4 "Head Installation" is Step 5 "Control Box Installation".".

2. *How does ChatGPT4 handle the extraction of a procedure across pages?* In the manufacturing domain, the instructions for "support plate installation" span two pages. This is example 2 handling the "Support plate installation" procedure in the set of procedures. The chatgpt response for the type 1 prompt to list all steps, substeps etc. accurately reflected the expected gold-standard. Thus we see that the agent can assimilate information across pages while maintaining the right context. As another example from the manufacturing domain is example 4 in the context of type 1 prompt for the "Removal and Installation of Mechanical Seal" sub-procedure for the "Shaft-seal maintenance" procedure. Here again, the chatgpt response aside from splitting some instruction types, is 90% in accordance to the gold-standard. It has successfully extracted the first step from the first page and the remaining steps from the next page of the manual. For the same instruction, when prompted in the 2-shot setting with examples of the desired response provided in the prompt, the chatgpt response exactly matches the gold-standard. This behaviour is consistently observed for other domains as well.

3. *Apart from text generation discrepancies, has ChatGPT4 completely overcome the limitation of LLMs of not being truly capable of mathematical logic or reasoning but simply still relegated as very powerful statistical text generators [2, 3]?* In the manufacturing domain, for type 3 comparison prompts, the language model was asked to compare the number of instructions given two contexts with procedures and reply which context had more instructions. Intriguingly, one of the contexts contained two procedures. Thus the task of the language model was to consider each independent procedure within each context and return which one had the most instructions. The incorrect model response over a relatively simple reasoning task offers further credence to the conjecture: are large language models simply very good statistical generators and otherwise incapable of truly reasoning?

4. *Are our instructions completely unambiguous to the model?* We observed that in some cases they might be ambiguous. For instance, in prompt 4, i.e. probing the model to produce nested instructions setting, our prompt reads as follows:

> Question: Can you provide a detailed list of the substeps of Step X in the given Context which refers to the "[name]" procedure? If there are no substeps, please reply with "no substeps". Answer:

Sometimes, if the main instruction itself is a rather long sentence, e.g., the generated step 7 here, the prompt above proves ambiguous where ChatGPT4 splits the long instruction into a sequence of steps as in this response. In-context learning alleviates ambiguity. In our 2-shot setting, the same prompt results with the correct response

5. *Can ChatGPT4 effectively extract information from manuals in a 2-column format, processing each column accurately?*

We find that it can extract content to create a response cleanly column-by-column. However, if queried about a procedure described in the first column, it may not be able to detect the end of the procedure as relegated just to that one column. It could continue generating text even including a new procedure starting in the second column of the same page. E.g., the following manual on operating tractors has two distinct procedures, i.e. Operating the Hydrostatic Transmission (listed completely in column 1) and Using Cruise Control - 1026R (listed completely in column 2). In the prompt 1 scenario, ChatGPT4 when asked to list steps for “Operating the Hydrostatic Transmission” (see example 1), it successfully extracts the relevant text but continues extracting text even for the “Cruise Control” procedure and lists its steps as substeps to the last step for “Operating the Hydrostatic Transmission.” See response here.

6. *Is ChatGPT4 able to comprehend the correct application of the ontology even though the generated response does not match the gold-standard?*

A step in the ontology is described as follows: first given an instance name and initialised as a Step type of a Plan. E.g., “kh-p-instance:Step2 a p-plan:Step ;” Next the step is assigned a label. E.g., “rdfs:label "Attach the hoses to the flowmeter ;”. Then the next step to the given step is specified. This can be a substep if the given step has substeps. E.g. “kh-p:nextStep kh-p-instance:SubStep2_1 ;”. Then the step in question is initialised as an instance of the corresponding main plan it belongs to. E.g., “kh-p:isStepOfPlan kh-p-instance:Plan1 ;”. For those steps with substeps, the name of a subplan is specified. E.g., “kh-p:isDecomposedAsPlan kh-p-instance:SubPlan2 .” This subplan will be the plan to which substeps of the main plan are initialised. For a complete example, see lines 48 to 52 in the gold-standard example 1 in the medical domain. Note if a step does not have substeps then the specification of the next step goes to the next main step and not the substep. In addition, the line specifying the decomposed plan of a step will not be present. Now looking at the ChatGPT4 response for the same procedure, and the same instruction step in lines 17 to 27, we see that it has incorrectly specified the step in turn leading to an incorrect application of the ontology. First as a next step, instead of specifying the sub step, it used the subplan. Then for the substeps instead of specifying the decomposed plan for the main step at the step specification, it specifies it at the plan level which was initialised as the next step of the main step leading to something meaningless and not machine-actionable.

An incorrect application of the ontology is also found in chatgpt response to procedure 3 “Installation of FM Type.” Specifically, take a look at the “Pole Assembly Installation” main step lines 12 to 23. The main step is initialised as a type of Plan. There is no specification of the decomposed plan with its substeps. Thus in the zero-shot setting, ChatGPT4 cannot be expected to correctly apply the ontology. Promising enough, this changes in the 2-shot setting, where ChatGPT4 shows two examples of the correct application of the ontology. Then via in-context learning, it is able to correctly apply the ontology. See the ChatGPT4 response in the 2-shot setting for the same example, as a perfect application of the ontology, thereby showing that ChatGPT4 can be successfully guided via in-context learning toward the correct application of an ontology. Thus in a sense, at least for the ontology setting, it appears necessary to query ChatGPT4 via the in-context learning methodology showing it some examples with the correct application of the ontology.

As an observational note, in simpler ontology application scenarios, i.e. when there are just steps with no substeps, it does very well. E.g. from agriculture, for example 4, the ChatGPT4 ontologised response for the prompt 1 scenario to list steps is almost identical to the gold-standard.

7. *Has ChatGPT4 hallucinated?* One needs to still be wary of the use of ChatGPT4 as it can still entirely hallucinate content. Consider the prompt 4 listing of nested procedures scenario, in the 2-shot setting, despite precise instructions as well as in-context example, when asked to list the substeps of step 3 for "installation of the FM type" procedure, ChatGPT4 still hallucinated the entire response. Compare this with the expected answer for substep3_1, 3_2, and 3_3.

8. *Is the 2-shot setting infallible, or does the model occasionally produce unexplained hallucinations?* In the 2-shot setting, we have also observed scenarios where the model has hallucinated text. While it may have grasped the ontology components and application relatively well, for reasons we found unexplainable the text generated as steps was entirely made up and could not be found in the manual. E.g., the generated ChatGPT4 response in the 2-shot setting compared with the ontologised gold-standard or text-based gold-standard.

4. Related Work

In prior research, it's vital to examine the methods used for procedural text mining and the incorporation of Large Language Models (LLMs) for knowledge extraction.

Knowledge Extraction from Unstructured Sources. Extracting complex knowledge from unstructured sources presents several challenges in several domains. This variability complicates the accurate extraction and structuring of relevant information through knowledge extraction algorithms which are usually applied to specific domains [4]. The intricate nature of these documents requires manual review by domain experts after automated extraction, underscoring the limitations of machine-learning-based approaches [5]. Innovative methods, including interactive dialogues and language models, have emerged to address the lack of readily available training data for machine learning methods [6, 7].

Procedural Text Mining and Large Language Models (LLMs). In response to the challenges mentioned earlier, our research delves into the field of procedural text mining, capitalizing on advancements in Natural Language Processing (NLP). Large Language Models (LLMs) have emerged as a pivotal tool in this endeavour, surpassing the capabilities of traditional symbolic AI and machine learning technologies [8, 9]. These models offer a means to address the intricate and extensive nature of procedural documents, with the potential to enhance knowledge extraction efficiency.

Integration of LLMs in Knowledge Extraction Large Language Models (LLMs) demonstrate exceptional capabilities in natural language processing, surpassing what conventional symbolic AI and machine learning technologies can achieve [10]. These capabilities have sparked a substantial increase in proofs of concept and practical applications of LLMs, suggesting their potential utility in various knowledge-related tasks [11]. Nevertheless, the exploration of methods for effectively integrating LLMs into structured, controllable, and repeatable approaches for the development and deployment of such applications in production is still in its early stages and requires further detailed consideration [12]. Similarly, our study centers on the integration of LLMs, notably the state-of-the-art GPT-4 model, in the context of extracting procedural knowledge from unstructured PDF documents.

Our solution proposes an innovative approach of combining large language models, specifically GPT-4, with in-context learning strategies to address the challenges of extracting procedural knowledge from unstructured PDF documents, offering a novel solution to the scarcity of training data in deep learning-based NLP techniques for procedure extraction.

5. Conclusion

In this study, we explored the feasibility of employing in-context learning with pre-trained language models to extract procedure elements from textual documents. We examined the native GPT-4 model and two customised variants, fine-tuned with procedure element definitions and limited examples. While GPT-4 is touted as the current most potent Large Language Model (LLM) with an expansive parameter count exceeding a trillion, its application is limited due to its proprietary nature. Notably, it also lacks the capacity for fine-tuning as of the time of writing this paper. In forthcoming research, we propose evaluating open-source LLMs, such as those related to T5 [13, 14, 15, 16] or Llama [17, 18]. This approach offers two key advantages: firstly, the adoption of open-source models would promote research democratization by removing paywalls as a barrier, and secondly, it facilitates potential enhancements, as these open models provide comprehensive technical insights into their pretraining datasets and strategies.

References

[1] P. Bellan, M. Dragoni, C. Ghidini, Process extraction from text: state of the art and challenges for the future, CoRR abs/2110.03754 (2021).

[2] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al., Emergent abilities of large language models, arXiv preprint arXiv:2206.07682 (2022).

[3] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al., Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, arXiv preprint arXiv:2206.04615 (2022).

[4] M. Y. Jaradeh, K. Singh, M. Stocker, A. Both, S. Auer, Better call the plumber: Orchestrating

dynamic information extraction pipelines, in: ICWE, volume 12706 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 240–254.

[5] Z. Dong, S. Paul, K. Tassenberg, G. Melton, H. Dong, Transformation from human-readable documents and archives in arc welding domain to machine-interpretable data, Comput. Ind. 128 (2021) 103439.

[6] A. Rebmann, H. van der Aa, Extracting semantic process information from the natural language in event logs, in: CAiSE, volume 12751, Springer, 2021, pp. 57–74.

[7] P. Bertoli, F. Corcoglioniti, C. D. Francescomarino, M. Dragoni, C. Ghidini, M. Pistore, Semantic modeling and analysis of complex data-aware processes and their executions, Expert Syst. Appl. 198 (2022) 116702.

[8] K. Sun, Y. E. Xu, H. Zha, Y. Liu, X. L. Dong, Head-to-tail: How knowledgeable are large language models (llm)? a.k.a. will llms replace knowledge graphs?, 2023. arXiv:2308.10168.

[9] P. Bellan, M. Dragoni, C. Ghidini, Extracting business process entities and relations from text using pre-trained language models and in-context learning, in: Enterprise Design, Operations, and Computing - 26th International Conference, EDOC 2022, Bozen-Bolzano, Italy, October 3-7, 2022, Proceedings, volume 13585 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 182–199. doi:10.1007/978-3-031-17604-3_11.

[10] K. Mahowald, A. A. Ivanova, I. A. Blank, N. Kanwisher, J. B. Tenenbaum, E. Fedorenko, Dissociating language and thought in large language models: a cognitive perspective, 2023. arXiv:2301.06627.

[11] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, D. Roth, Recent advances in natural language processing via large pre-trained language models: A survey, ACM Comput. Surv. (2023). URL: https://doi.org/10.1145/3605943. doi:10.1145/3605943.

[12] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, 2023. arXiv:2306.08302.

[13] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, The Journal of Machine Learning Research 21 (2020) 5485–5551.

[14] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, Q. V. Le, Finetuned language models are zero-shot learners, arXiv preprint arXiv:2109.01652 (2021).

[15] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, arXiv preprint arXiv:2210.11416 (2022).

[16] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, et al., The flan collection: Designing data and methods for effective instruction tuning, arXiv preprint arXiv:2301.13688 (2023).

[17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).