Computational Models of Argument
C. Reed et al. (Eds.)

doi:10.3233/FAIA240311

"I'd Like to Have an Argument, Please": Argumentative Reasoning in Large Language Models

Adrian DE WYNTER [a,b,1] and Tangming YUAN [b]
[a] *Microsoft*
[b] *The University of York*
ORCiD ID: Adrian de Wynter https://orcid.org/0000-0003-2679-7241, Tangming Yuan https://orcid.org/0000-0003-1697-7003

Abstract. We evaluate two large language models (LLMs) ability to perform argumentative reasoning. We experiment with argument mining (AM) and argument pair extraction (APE), and evaluate the LLMs' ability to recognize arguments under progressively more abstract input and output (I/O) representations (e.g., arbitrary label sets, graphs, etc.). Unlike the well-known evaluation of prompt phrasings, abstraction evaluation retains the prompt's phrasing but tests reasoning capabilities. We find that scoring-wise the LLMs match or surpass the SOTA in AM and APE, and under certain I/O abstractions LLMs perform well, even beating chain-of-thought–we call this *symbolic prompting*. However, statistical analysis on the LLMs outputs when subject to small, yet still human-readable, alterations in the I/O representations (e.g., asking for BIO tags as opposed to line numbers) showed that the models are not performing reasoning. This suggests that LLM applications to some tasks, such as data labelling and paper reviewing, must be done with care.

Keywords. GPT-4, GPT-3, reasoning, argument mining, argument pair extraction

1. Introduction

Large language models (LLMs) such as GPT-4 [1] have shown to have spectacular accuracy on a variety of tasks. Hence, attempts have been made to automate more complex tasks reliant on argumentative reasoning, such as data labelling [2] and scientific paper reviews [3]. Argumentative reasoning encompasses formal and informal logic, and requires a deep understanding of, and reasoning over, the pragmatic context. Hence, to understand the reliability of LLMs in these tasks we must also evaluate their argumentative reasoning capabilities. This goes beyond determining whether the model can generate relevant responses, and asks if it can robustly reason over the context and solve the task.

We evaluate the argumentative reasoning capabilities of two LLMs, GPT-3 [4] and GPT-4 in two tasks, argument mining (AM) and argument pair extraction (APE [5]).[2] We do this by an *abstraction evaluation*: measuring progressively more abstract input

[1] Corresponding Author: Adrian de Wynter, adewynter@microsoft.com
[2] Prompts, code, and outputs are in https://github.com/adewynter/argumentation-llms

and output (I/O) representations. Unlike prompt-phrasing evaluations, to which LLMs are known to be sensitive [6, 7], our evaluation maintains the task description untouched, and only alters the signature of the data. For example, adding line numbers to the input and requesting the output to be from a specific set (e.g., $\{0, 1\}$) is a conceptually minor I/O representation change. This retains the task description, but requires some level of reasoning to return a correct and parseable response given the specified signatures. In this paper we then measure the LLMs' argumentative reasoning capabilities indirectly, by testing their ability to *robustly* recognize arguments when altering the I/O.

1.1. Findings

In terms of raw scoring, we find that GPT-4 is able to reach SOTA performance in APE, and near-SOTA in AM. However, our analysis shows that:

1. LLM scoring varies dramatically with the abstraction level, which suggests a lack of comprehension of the task. Note that across our experiments the task description remained fixed, and we only altered the I/O representation.
2. *Symbolic prompting* (that is, low-abstraction, hints-enabled inputs) bests other approaches, including chain-of-thought (CoT) [8].
3. CoT approaches are robust to abstraction, but its output distributions over abstractions are similar, which we attribute to its templatized nature.
4. LLM scores worsen with more exemplars, indicating poor inductive reasoning capabilities.

We conclude that the LLMs are unable to reason reliably in an argumentative setting.

2. Related Work

We discuss evaluations of LLMs as it pertains to argumentative reasoning. For non-LLM-based approaches see [9]; and for a survey on reasoning in LLMs see [10]. LLMs are relatively new, though they typically outperform non-LLM approaches, for example in AM [11] and argument evaluation [12, 13], in terms of raw scoring. Indeed, LLMs have been observed to generalize to (read: score well in) unseen tasks without training, thus raising the question as to whether they can reason about the prompts; or are just regurgitating their training data or returning semantically-close responses. To some, this generalization is an indication of emergent reasoning capabilities [14, 15]; but it has been posed that with better statistics this evidence of emergence disappears [16]. Some tests, such as GPT-4's own technical report [1], tout remarkable reasoning capabilities. There is also evidence to the contrary, e.g. in code generation [17], scientific questions [18], first-order logic under fictional worlds [19], and arithmetic [20]; more generally, tasks with significant reasoning depths cause LLMs to fail [21, 22]. It has hence been suggested that LLMs do not actually reason, but rely on heuristics (e.g., semantic similarity) [23]. Remark that these studies are limited to the prompts and versions of the models available then. It was suggested that LLMs are not meant for formal reasoning, and it is better to evaluate them in real-world (informal, inductive) scenarios [22]; yet GPT-3 cannot mimic human-like inductive reasoning [24], or understand the prompts [7]. LLMs have also been found to not be competent in legal reasoning, due to their inability to make good arguments [25]. LLMs may also retrieve dialogue acts, in line with their success as chatbots, but do not understand offers in negotiations (i.e., pragmatics) [26].

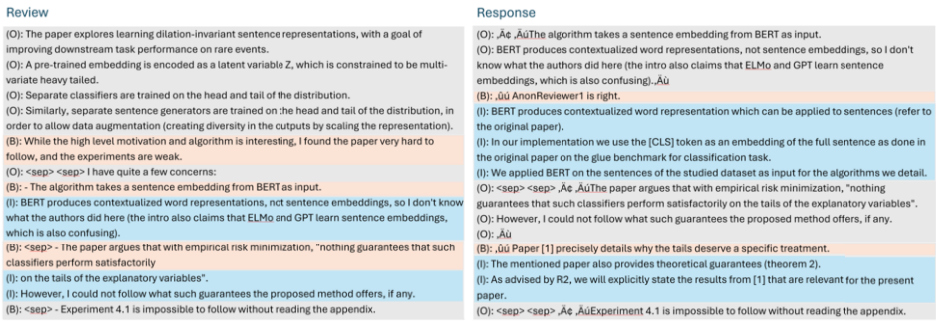


Figure 1. Examples of the entries in RRv2. The BIO tag is in parenthesis. Highlighted in orange and blue are the "B" and "I" labels corresponding to an argument. In AM, review and rebuttal (response) passages are independent, and the task is to assign BIO tags to every line. In APE, the task is to match arguments from the review to their corresponding responses. In the above the first argument is unmatched, and the second ("BERT produces...") pairs to the first argument from the response.

3. Methodology

3.1. Data

Throughout this paper, we utilize the Review-Rebuttal Submission-v2 (RRv2) dataset [5]. It is a comprehensive corpus focused on long-distance relationships between statements, and includes both AM and APE. It has 4,764 (474 for test) pairs of review and rebuttal passages related to scientific article submissions. Each passage is sentence-separated, and includes multiple arguments. It is human-labelled. For AM, each sentence is labelled with a BIO tag,[3] and the model must retrieve (label) each sentence from the review and rebuttal entries. In AM the distinction between review and rebuttal is irrelevant: each entry is treated as a separate point in the corpus. For APE, the task is to align the arguments within each review-rebuttal pair: every argument made by a reviewer must be mapped, when applicable, to a response from the rebuttal. This is normally represented as a binary matrix with overlaps [27, 28]. Prior to use we clean the text from tag and sentence delimiters. See Figure 1 for a sample of the corpus.

3.2. LLMs Evaluated

We evaluate GPT-4 and the TEXT-DAVINCI-003 variant of GPT-3 ("GPT-3.51"). Both models are autoregressive language models, instruction-pretrained [29, 30] and tuned with reinforcement learning with human feedback [30, 31]. For GPT-4, there are no details released around the architecture, model size, or training data. It is considered better than GPT-3 at more complex tasks [1]. The variant of GPT-4 we used ("GPT-4-0613") has a context length of 32,768 tokens; GPT-3 has 4,097 tokens.[4]. Throughout our experiments, for both LLMs we set the temperature to 0.8, the maximum return tokens based on the task, and left everything else as default. To account for randomness, we report the average of five calls per point to the Azure OpenAI API.

[3] In the RRv2 corpus, the BIO tags correspond to the Beginning, Inner, and Outer parts of an argument.
[4] https://platform.openai.com/docs/models

3.3. Prompting

Prior to starting the work we tuned the prompt phrasing for best performance. When using n exemplars, we used the first n points from the development set. For AM we prompted GPT-4 with and without CoT. CoT conditions the LLM to work step-by-step by following a templatized process (e.g., "Let's think about this step-by-step..."). It is known to provide good results in multiple reasoning tasks [8, 15, 32]. All our prompts followed the structure from GPT-4's technical report [1], as we observed it produced more reliable outputs. For CoT we followed a tuned template ("Let's read line-by-line and solve this step by step") indicating which sentence was being read, as well as the rationale. For example, "We now read {SENTENCE}. It follows the previous argument, and hence it is labelled with an 'I'." Sample prompts can be found in Figures 2a and 2b.

RRv2 is measured with binary F_1 (F_{01}) for APE, and micro-F_1 (F_μ) for AM [5]. Our prompts have specified a return format to signal the beginning of parsing.

3.4. Baselines

Our baselines are the MLMC [28] and MRC-APE [27] models. MLMC approaches APE as a table-filling problem: passages are related by their pairing on a table and it relies on an especially designed encoding scheme and loss. MRC-APE phrases it as a reading comprehension task: first, the model does AM, and then pairs the detected arguments. This approach is effective when using longer-context layers, of up to $4,096$ tokens. We additionally consider random guessers for AM (around 33% F_μ) and APE (14% F_{01}).

3.5. Settings

We have named our settings (i.e., representations) as *concrete* and *symbolic*, to distinguish the approach taken towards representing the task. Concrete returns full sentences, while symbolic encompasses a variety of I/O symbols. This is only for practical purposes: symbolic approaches cover multiple I/O representations, some of which may be easier than concrete; and, strictly speaking, the concrete setting is a type of symbolic representation [33]. See Table 1 for a full description of the settings tested.

The *concrete* setting we instruct the LLM to return lines in text based on the prompt: in AM, it must be part of an argument, in APE, an argument pair. To distinguish the "B" and "I" labels, we enforce a specific return format to work with our parsing code via a special token (`|START|`). For scoring concrete settings we expect an exact text match.

In *symbolic* settings the LLM must return symbols (labels) based on the prompt. This requires more reasoning steps than in concrete settings: the LLM is solving AM *and* labelling the span with an arbitrary label set defined in the prompt. In AM symbolic we evaluated two types of labels: BIO tags and line indices. For APE we evaluated line indices and the full binary matrix representation. We also used abstract meaning representation (AMR [34]) graphs, which are used in argument interpretation [35].

4. Experiments and Results

We report our results comparing raw scores with respect to our settings (Section 4.1); number of exemplars (Section 4.2); and I/O representations (Section 4.3).

```
Extract from the passage all the arguments.
The output should be in the form:
|begin response|
|START| line from argument 1
line from argument 1
|START| line from argument 2
line from argument 2
...
etc
|end response|
For example,
{EXEMPLARS GO HERE}
Passage:
{PASSAGE GOES HERE}
Response:
|begin response|
```

(a) Sample concrete AM prompt. The model must mark every new argument with a special token ("|START|") for identification. In APE we ask for the pairing (e.g., "return all arguments from the response that match those of the review").

```
Extract from the passage all the arguments.
Label the beginning of every argument with a
"B". Label the rest of the argument with an
"I". Label every line that is not part of an
argument with an "O".
The output should be in the form:
|begin response|
B
I
...
etc
|end response|
For example,
{EXEMPLARS GO HERE}
Passage:
{PASSAGE GOES HERE}
Response:
|begin response|
```

(b) Sample symbolic AM prompt. The model must return BIO tags. In other representations (e.g., indices) the model must to mark the B-label in parenthesis (e.g, "(15) 16 17"). In APE this output is of the form "argument lines: response lines", and we convert into a binary matrix for scoring.

Figure 2. Sample prompts for our concrete (left) and symbolic (right) settings. Exemplars, if any, are included in the prompt. For zero-shot we only specify the output representation. Note how the actual task definition and prompt structure remains unchanged, and we only alter the I/O representations.

Task	Input	Output (label set)	Abstraction
AM*	Text	Text and |START|	Lowest
AM*	Text with indices	Indices	Low
AM*	Text	Indices	Medium
AM*	Text	BIO tags	Medium-high
AM	Text with AMR graph	BIO tags	High
AM	AMR graph	BIO tags	Highest

Table 1. Input representations tested, in roughly increasing order of abstraction. Tasks marked with an asterisk (*) were tested with and without CoT. The first row is our concrete reasoning setting. Our ranking of abstraction is arbitrary: we consider the text with indices marked inline less abstract than a text without them, since the former provides a "hint" of what the label set is supposed to be like. Output representations with BIO tags as the output are more abstract, since it requires rule matching to determine the labeling.

4.1. AM/APE: Symbolic and Concrete Reasoning

Results for the best-performing prompts and settings are in Table 2, and a description of every setting in Table 1. In AM the LLMs did well but did not beat the SOTA. The best-performing symbolic setting had line indices included in the input representation, and requested the indices of each argument as the output representation. To convert to BIO tags, we instructed the model to return the "B" labels as indices enclosed in parentheses. Not including the indices in the input did not lead to an equivalently good performance.

For APE, GPT-4 consistently bested the best-performing models (+14% F_{01}). Both symbolic and concrete approaches did well with respect to the existing non-LLM-based baselines. We tested other approaches, such as first extracting the arguments and then matching them, but it did not yield sufficiently good results. Requesting a binary matrix output led to extremely poor performance (9.88% F_{01}, below random). Due to token-length and budget limitations, we were unable to test CoT and AMR in APE.

CoT approaches had generally better performance than their non-CoT counterparts, even though they use fewer exemplars. The only exception to this was symbolic prompting (indices inline and indices as output) where the difference was 3% points.

Model	AM F_μ	APE F_{01}
GPT-3 (concrete)	39.86 ± 0.51	18.58 ± 0.70
GPT-4 (concrete)	64.51 ± 0.53	53.84 ± 0.73
GPT-3 (symbolic)	62.00 ± 0.32	20.15 ± 0.91
GPT-4 (symbolic)	70.63 ± 0.21	49.85 ± 0.96
MRC-APE	72.43	39.92
MLMC	71.35	32.81

Table 2. Results for the AM and APE tasks in our settings. The best-performing symbolic prompt had indices inline and indices as the output. We also report MLMC and MRC-APE, the two best-performing, non-LLM-based approaches for RRv2. GPT-4 almost matched the existing baselines in AM and bested them in APE.

4.2. Performance and Number of Exemplars

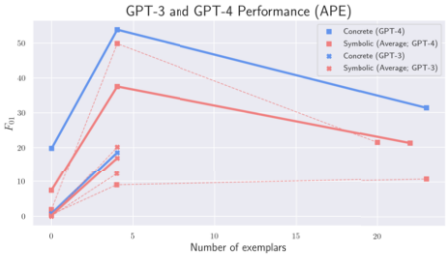
We compared the number of exemplars ($\{0, 4, 8, 16, \tau\}$, where τ is the maximum number) with the LLM performance. For GPT-3, τ tended to be around 4; for GPT-4 it varied, with an average of 44 for symbolic, non-CoT approaches in AM (15 CoT, 9 AMR) and 29 concrete; and 22 for both approaches in APE. Results are in Figures 3a and 3b. The LLMs peaked in performance at 4 exemplars, and their scores decreased from there. This was independent of the task and setting. We did not observe this trend in CoT.

4.3. Performance and Input Representation

In this section we focused on GPT-4 and AM and the following representations: text with indices inline, plain text, and with and without an AMR graph. There is no rigorous way to quantify the level of abstraction for these. However, we consider the concrete approach to be least abstract; "hints" (indices inline, indices in output) to be slightly more abstract; and purely symbolic input representations (AMR graphs) as most abstract. Other I/O representations are ranked based on the output representation: BIO tags are more abstract than indices (they require rules for matching); and both are more complex than concrete settings, since outputting a matching string is easier than mapping to an arbitrary symbol. The list of experiments is in Table 1, and our results in Figure 4.

For non-CoT approaches, we observed noticeable improvements in low-abstraction scenarios (indices inline and indices in output; concrete). As it rose, LLMs scored worse, though remaining above random. The results are significant under a Welch's t-test on the

(a) Number of exemplars versus F_μ in AM. CoT (thin solid red lines) outperformed concrete and all but one of the symbolic approaches.



(b) Number of exemplars versus F_{01} in APE. We could not test CoT and AMR due to limitations on token length and budget.

Figure 3. Exemplar number versus score for AM and APE. In blue is the concrete approach; thick red line is the averaged performance (symbolic; dashed red lines). Last entry in all plots is the average maximum number of exemplars supported. LLM scoring in non-CoT peaked at around 4 exemplars, and decreased afterwards.

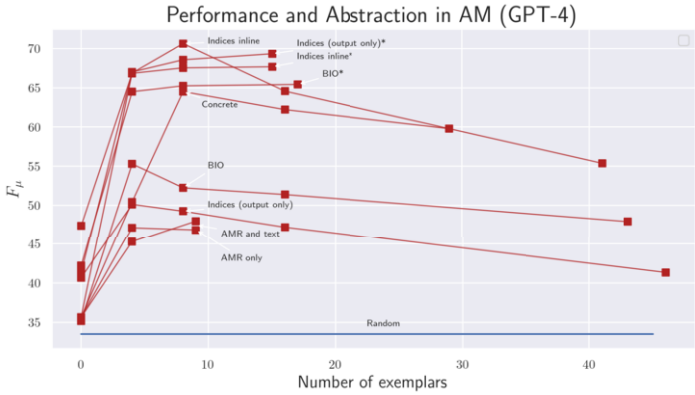


Figure 4. Effect of abstraction in the I/O representation with respect to F_μ in AM. For non-CoT, the more abstract the input, the more difficult it is for the model to solve the task. Even the most abstract representations (AMR graphs) are noticeably better than a random. In CoT (top, marked with an asterisk) consistently outperformed non-CoT, even when the maximum number of exemplars supported is much lower, and regardless of abstraction level. We did not test CoT with AMR in AM due to token limitations.

prediction arrays. When the input text is unaltered, line numbers or BIO tags make no difference in predictions with CoT ($p \approx 0.77$; large p-values imply the distributions have identical expected values), but are noticeably worse without ($p < 0.05$). Another t-test shows that $t < -0.86$ and $p \approx 0.39$ for inline indices when compared with its non-CoT version. Hence we reject the null hypothesis that the distributions are distinct. Since the performance in non-CoT was better on average (67% vs 70% F_μ), it is possible that CoT harms performance on outlier points. We compared CoT in other scenarios, and also observed large p values when comparing with only indices in the output (< 0.98); and when comparing the latter with BIO tags ($p < 0.77$). This suggests that CoT is effective in highly abstract scenarios, such as requesting BIO tags, but detrimental otherwise.

5. Discussion

The LLMs scored highly in AM and APE, to the point of beating or almost matching the existing SOTA models. This is not sufficient to claim that the models are able to perform argumentative reasoning. When we altered the I/O representation with conceptually minor changes–such as adding line numbers or requesting a BIO set as opposed to integers–the LLMs had noticeably different performances. An explanation for the relatively low scores in APE is that the LLMs failed to generalize due to the length of the task; which causes problems in transformer-based models [21].

CoT prompts had *on average* higher scores than their non-CoT counterparts. They also yielded better scores in ill-posed (overly abstract) scenarios. However, our analysis showed that the output distributions for all CoT approaches were rather similar. This is perhaps indicative that CoT allows the models to return the same output regardless of representation. This appears to be due to the templatized nature of CoT, i.e., "Let's read line-by-line and solve step-by-step", and the specific steps needed to generate the output regardless of the I/O representation. This itself is not indicative of reasoning.

Finally, there is a clear peak at four exemplars with respect to the model's downstream performance. This suggests that, assuming that more exemplars imply better information about the task, the LLMs are not accurately performing inference from the data provided. This exemplar effect did not extend to CoT settings, though we do not discard the possibility that models with longer token limitations could also show this trend. Overall, we pose that the models are unable to reason in an argumentative setting, but their scores give an excellent appearance of being able to do so.

6. Limitations

Our analysis has three main limitations. In terms of reasoning evaluation, it could be argued that our results are not complete in terms of evaluating argumentative reasoning capabilities. We agree: *recognizing* an argument is not the same as *deciding* its quality. However, without the ability of the model to show that it is able to recognize arguments and identify relations between them, any potentially generated argument or result evaluating the model's performance in these tasks is untrustworthy.

We factored out, to an extent, potential data contamination, which is known to impact downstream model performance [36, 37], by tasking the model to recognize arguments from the passage. However, we are unable to guarantee that the models have not been trained with this data, and therefore have at least some bias towards these results. Finally, we only evaluated two models, so our results may not extend to other LLMs. Likewise, we did not fine-tune the models, and opted instead to treating them as generalists performing in-context learning, in line with their contemporary usage.

7. Conclusion

We evaluated the argumentative reasoning capabilities of GPT-3 and GPT-4, by measuring whether they could recognize arguments from a passage–the first step on performing such reasoning. The LLMs score well in AM and APE, beating or nearly-matching the

SOTA. However, statistical analysis on the LLMs' predictions when subject to small, yet still human-readable, alterations in the I/O representations showed that the LLMs were extremely sensitive to the abstraction level and the number of exemplars. Hence, we concluded that they were not reasoning over the arguments seen.

However, symbolic prompting strategies (e.g. reducing the abstraction level of the prompt by adding line numbers) allowed the LLMs to score well and even beat CoT. We were also unable to conclude that CoT helped argumentative reasoning in LLMs, but did observe more robust results due to its templatized nature. Our analysis implies that it helps mitigate issues stemming from overly abstract or ill-conditioned problems.

As mentioned in Section 6, we did not evaluate the LLMs' ability to judge an argument's strength, or to provide reasonable rebuttals. Moreover, due to experimental limitations, we were unable to evaluate AMR, the most abstract setting we tested, with CoT, and other prompting strategies, such as Tree-of-Thoughts [38]. This, along with further evaluation on benchmarks specific to reasoning, could provide valuable insights on to what extent these models are able to discern abstract input representations. Further work could explore these issues. Overall, our work suggests that LLM usage in areas like data labelling and paper reviewing must be exercised with care and good judgement.

Acknowledgements

The authors wish to thank Liying Cheng for answering questions about the RRv2 dataset.

References

[1] Open AI. GPT-4 Technical Report. Open AI; 2023. Available from: https://arxiv.org/abs/2303.08774v2.

[2] Cheng L, Li X, Bing L. Is GPT-4 a Good Data Analyst? ArXiv. 2023;abs/2305.15038. Available from: https://arxiv.org/abs/2305.15038.

[3] Liu R, Shah NB. ReviewerGPT? An Exploratory Study on Using Large Language Models for Paper Reviewing. ArXiv. 2023;abs/2306.00622. Available from: https://arxiv.org/abs/2306.00622.

[4] Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language Models Are Few-Shot Learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NeurIPS'20. Red Hook, NY, USA: Curran Associates Inc.; 2020. .

[5] Cheng L, Bing L, Yu Q, Lu W, Si L. APE: Argument Pair Extraction from Peer Review and Rebuttal via Multi-task Learning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics; 2020. p. 7000-11. Available from: https://aclanthology.org/2020.emnlp-main.569.

[6] Lu Y, Bartolo M, Moore A, Riedel S, Stenetorp P. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Dublin, Ireland: Association for Computational Linguistics; 2022. p. 8086-98. Available from: https://aclanthology.org/2022.acl-long.556.

[7] Webson A, Pavlick E. Do Prompt-Based Models Really Understand the Meaning of Their Prompts? In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics; 2022. p. 2300-44. Available from: https://aclanthology.org/2022.naacl-main.167.

[8] Wei J, Wang X, Schuurmans D, Bosma M, brian ichter, Xia F, et al. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In: Oh AH, Agarwal A, Belgrave D, Cho K, editors. Advances in Neural Information Processing Systems; 2022. Available from: https://openreview.net/forum?id=_VjQlMeSB_J.

[9] Lawrence J, Reed C. Argument Mining: A Survey. Computational Linguistics. 2020 01;45(4):765-818. Available from: https://doi.org/10.1162/coli_a_00364.

[10] Huang J, Chang KCC. Towards Reasoning in Large Language Models: A Survey. In: Rogers A, Boyd-Graber J, Okazaki N, editors. Findings of the Association for Computational Linguistics: ACL 2023. Toronto, Canada: Association for Computational Linguistics; 2023. p. 1049-65. Available from: https://aclanthology.org/2023.findings-acl.67.

[11] Zhao F, Yu F, Trull T, Shang Y. A New Method Using LLMs for Keypoints Generation in Qualitative Data Analysis. In: 2023 IEEE Conference on Artificial Intelligence (CAI); 2023. p. 333-4.

[12] Van der Meer M, Reuver M, Khurana U, Krause L, Baez Santamaria S. Will It Blend? Mixing Training Paradigms & Prompting for Argument Quality Prediction. In: Proceedings of the 9th Workshop on Argument Mining. International Conference on Computational Linguistics; 2022. p. 95-103. Available from: https://aclanthology.org/2022.argmining-1.8.

[13] Holtermann C, Lauscher A, Ponzetto S. Fair and Argumentative Language Modeling for Computational Argumentation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Dublin, Ireland: Association for Computational Linguistics; 2022. p. 7841-61. Available from: https://aclanthology.org/2022.acl-long.541.

[14] Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, et al. Emergent Abilities of Large Language Models. Transactions on Machine Learning Research. 2022. Survey Certification. Available from: https://openreview.net/forum?id=yzkSU5zdwD.

[15] Suzgun M, Scales N, Schärli N, Gehrmann S, Tay Y, Chung HW, et al. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In: Findings of the Association for Computational Linguistics: ACL 2023. Toronto, Canada: Association for Computational Linguistics; 2023. p. 13003-51. Available from: https://aclanthology.org/2023.findings-acl.824.

[16] Schaeffer R, Miranda B, Koyejo S. Are Emergent Abilities of Large Language Models a Mirage? In: Thirty-seventh Conference on Neural Information Processing Systems; 2023. Available from: https://openreview.net/forum?id=ITw9edRDlD.

[17] Liu J, Xia CS, Wang Y, Zhang L. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In: Thirty-seventh Conference on Neural Information Processing Systems; 2023. Available from: https://openreview.net/forum?id=1qvx610Cu7.

[18] Schulze Balhorn L, Weber S Jana M Buijsman, Hildebrandt JR, Ziefle M, Schweidtmann AM. Empirical assessment of ChatGPT's answering capabilities in natural science and engineering. Nature. 2024;14.

[19] Saparov A, He H. Language Models are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought. In: International Conference on Learning Representations (ICLR); 2023. .

[20] Dziri N, Lu X, Sclar M, Li XL, Jian L, Lin BY, et al. Faith and Fate: Limits of Transformers on Compositionality. ArXiv. 2023;abs/2305.18654.

[21] Anil C, Wu Y, Andreassen AJ, Lewkowycz A, Misra V, Ramasesh VV, et al. Exploring Length Generalization in Large Language Models. In: Oh AH, Agarwal A, Belgrave D, Cho K, editors. Advances in Neural Information Processing Systems; 2022. Available from: https://openreview.net/forum?id=zSkYVeX7bC4.

[22] Valmeekam K, Olmo A, Sreedharan S, Kambhampati S. Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change). In: NeurIPS 2022 Foundation Models for Decision Making Workshop; 2022. Available from: https://openreview.net/forum?id=wUU-7XTL5XO.

[23] Patel A, Bhattamishra S, Goyal N. Are NLP Models really able to Solve Simple Math Word Problems? In: Toutanova K, Rumshisky A, Zettlemoyer L, Hakkani-Tur D, Beltagy I, Bethard S, et al., editors. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Online: Association for Computational Linguistics; 2021. p. 2080-94. Available from: https://aclanthology.org/2021.naacl-main.168.

[24] Han SJ, Ransom KJ, Perfors A, Kemp C. Human-like property induction is a challenge for large language models. In: Proceedings of the 44th Annual Conference of the Cognitive Science Society (CogSci 2022). Toronto, Canada: Annual Conference of the Cognitive Science Society (CogSci); 2022. p. 1-8.

[25] Nguyen HT, Fungwacharakorn W, Satoh K. Enhancing Logical Reasoning in Large Language Models to Facilitate Legal Applications. ArXiv. 2023;abs/2311.13095. Available from: https://arxiv.org/abs/2311.13095.

[26] Lin E, Hale J, Gratch J. Toward a Better Understanding of the Emotional Dynamics of Negotiation with Large Language Models. In: Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing. Washington DC USA: ACM; 2023. p. 545–550. Available from: https://dl.acm.org/doi/10.1145/3565287.3617637.

[27] Bao J, Sun J, Zhu Q, Xu R. Have my arguments been replied to? Argument Pair Extraction as Machine Reading Comprehension. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Dublin, Ireland: Association for Computational Linguistics; 2022. p. 29-35. Available from: https://aclanthology.org/2022.acl-short.4.

[28] Cheng L, Wu T, Bing L, Si L. Argument Pair Extraction via Attention-guided Multi-Layer Multi-Cross Encoding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics; 2021. p. 6341-53. Available from: https://aclanthology.org/2021.acl-long.496.

[29] Wei J, Bosma M, Zhao V, Guu K, Yu AW, Lester B, et al. Finetuned Language Models are Zero-Shot Learners. In: International Conference on Learning Representations; 2022. Available from: https://openreview.net/forum?id=gEZrGCozdqR.

[30] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, et al. Training language models to follow instructions with human feedback. In: Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, editors. Advances in Neural Information Processing Systems. vol. 35. Curran Associates, Inc.; 2022. p. 27730-44. Available from: https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[31] Christiano PF, Leike J, Brown T, Martic M, Legg S, Amodei D. Deep Reinforcement Learning from Human Preferences. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017. Available from: https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[32] Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y. Large Language Models are Zero-Shot Reasoners. In: Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, editors. Advances in Neural Information Processing Systems. vol. 35. Curran Associates, Inc.; 2022. p. 22199-213. Available from: https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

[33] Brachman RJ, Levesque HJ. Knowledge Representation and Reasoning. Morgan-Kaufmann; 2004.

[34] Banarescu L, Bonial C, Cai S, Georgescu M, Griffitt K, Hermjakob U, et al. Abstract Meaning Representation for Sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. Sofia, Bulgaria: Association for Computational Linguistics; 2013. p. 178-86. Available from: https://aclanthology.org/W13-2322.

[35] Opitz J, Heinisch P, Wiesenbach P, Cimiano P, Frank A. Explainable Unsupervised Argument Similarity Rating with Abstract Meaning Representation and Conclusion Generation. In: Proceedings of the 8th Workshop on Argument Mining. Punta Cana, Dominican Republic: Association for Computational Linguistics; 2021. p. 24-35. Available from: https://aclanthology.org/2021.argmining-1.3.

[36] Lee J, Le T, Chen J, Lee D. Do Language Models Plagiarize? In: Proceedings of the ACM Web Conference 2023. WWW '23. New York, NY, USA: Association for Computing Machinery; 2023. p. 3637–3647. Available from: https://doi.org/10.1145/3543507.3583199.

[37] De Wynter A, Wang X, Sokolov A, Gu Q, Chen SQ. An evaluation on large language model outputs: Discourse and memorization. Natural Language Processing Journal. 2023;4:100024. Available from: https://www.sciencedirect.com/science/article/pii/S2949719123000213.

[38] Yao S, Yu D, Zhao J, Shafran I, Griffiths TL, Cao Y, et al. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In: Thirty-seventh Conference on Neural Information Processing Systems; 2023. Available from: https://openreview.net/forum?id=5Xc1ecxO1h.