

An Overview of Using Large Language Models for the Symbol Grounding Task in ABC Repair System

Pak Yin Chan¹, Xue Li¹ and Alan Bundy¹

¹*School of Informatics, The University of Edinburgh, United Kingdom*

Abstract

The ABC Theory Repair System (ABC) has demonstrated its success in facilitating users to repair faulty theories utilizing distinct techniques. Yet, comprehending ABC-repaired theories becomes more challenging due to the presence of dummy constants or predicates introduced by ABC. In this paper, we propose a grounding system by incorporating Large Language Models (LLMs) to provide these dummy items with meaningful names. By applying ABC and grounding alternately, the resulting theory is both fault-free and semantically meaningful. Moreover, our study shows that LLMs without fine-tuning still exhibit capabilities of common knowledge, and their grounding performances are enhanced by providing sufficient background or asking for more returns.

Keywords

Large language model, Closed-book question answering, Faulty logical theory repair, Automated theorem proving

1. Introduction

Logical theory stands as a reasoning tool in the field of artificial intelligence (AI), representing structured and precise representations of relationships among objects [1]. The theory needs to be refined to cope with new observations [2]. When users introduce novel information, the original theory may either make incorrect predictions or fail to predict the expected truth. [3].

To address faults in flawed logical theories, the ABC Theory Repair System (ABC) was proposed, automatically to integrate several repair techniques like abduction [4], belief revision [5], and conceptual change with reformation [6] based on user observations [2, 3]. Although ABC is capable of generating error-free theories, users might encounter confusion due to the appearance of dummy constants or predicates (hereafter “dummy items”) in these theories. Example 1 illustrates a repaired theory featuring dummy items. We can prove Camilla is equal to Diana in the original theory, which is incompatible with the fact that they are not the same person. To repair this theory, ABC introduces two dummy constants, “dummyConst1” and “dummyConst2”, to differentiate between two types of mothers.

These dummy items come into existence when ABC employs repair plans involving the reformation technique. They are assigned names prefixed with “dummy” as their meanings are unknown to ABC [3]. Presently, users are responsible for manually assigning names to these


Cognitive AI 2023, 13th-15th November, 2023, Bari, Italy.

✉ s2341572@ed.ac.uk (P. Y. Chan); xue.shirley.li@ed.ac.uk (X. Li); A.Bundy@ed.ac.uk (A. Bundy)

🆔 0009-0002-9631-5543 (P. Y. Chan); 0000-0002-6665-2242 (X. Li); 0000-0002-0578-6474 (A. Bundy)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

dummy items. In the previous example, users can deduce that “dummyConst1” and “dummyConst2” are referring to a birth mother and a stepmother respectively. Yet, there are instances when users face uncertainty in naming these items. Since the current implementation of ABC does not encompass the consideration of their semantic implications, numerous generated repaired theories might be logically consistent but devoid of semantic meaning.

Example 1 : A Comparison of Original and Repaired Motherhood Theory	
Original Theory:	
	$mum(X, Z) \wedge mum(Y, Z) \implies X = Y$ $\implies mum(camilla, william)$ $\implies mum(diana, william)$
<hr/>	
Repaired Theory:	
	$mum(X, Z, dummyConst1) \wedge mum(Y, Z, dummyConst1) \implies X = Y$ $\implies mum(camilla, william, dummyConst2)$ $\implies mum(diana, william, dummyConst1)$

The challenge of attributing meanings to meaningless symbols is known as the “symbol grounding problem”, an important problem in Cognitive Science [7]. To tackle this problem automatically, we require tools with access to common-sense knowledge, enabling them to suggest potential names for users to consider. Although Large Language Models (LLMs) exhibit inconsistencies in reasoning [8, 9], studies indicate that they store extensive relational knowledge of the training datasets during pretraining [10, 11, 12]. This suggests that we can leverage LLMs to propose possible meanings for dummy items by presenting propositions involving these items in natural language.

Our study demonstrates an application of LLMs to solve the symbol grounding problem. The primary objective of this paper is *to enrich the semantic content of the repaired theory by utilizing LLMs to replace the names of dummy items with semantically meaningful content*. We hypothesize that the closed-book question answering (CBQA) task [12] with LLMs helps to conduct the symbol grounding challenge within ABC, which in the CBQA task, LLMs generate responses based on their training data solely, without having access to external sources [11, 12]. To explore how well can LLMs provide meanings of dummy items in the repaired theory by the ABC, as a way of enhancing the semantics of the repaired theory, we propose a system of symbol grounding for ABC to determine the meanings of dummy items that involve user interactivity.

2. LLM-grounding system

Figure 1 illustrates the process employed by the grounding system. We first parse the input theory in Datalog, a subset of First Order Logic (FOL) [13], and the system sets up records of constants and predicates (G1). These records facilitate the detection of ungrounded dummy items. For each dummy item, the system interprets the associated axioms into a natural language

question (G2). After grounding with the LLM that users choose (G3), the system presents all available choices by that LLM. Users can choose the LLM-suggested answers or suggest new answers by themselves (G4a). After that, the system also recommends users use the existing theory items with high similarity to any previously suggested answers (G4b). Once all dummy items are successfully grounded, the system proceeds to replace these items with the selected answers (G5) and exports all possible grounded theories in Datalog [13].

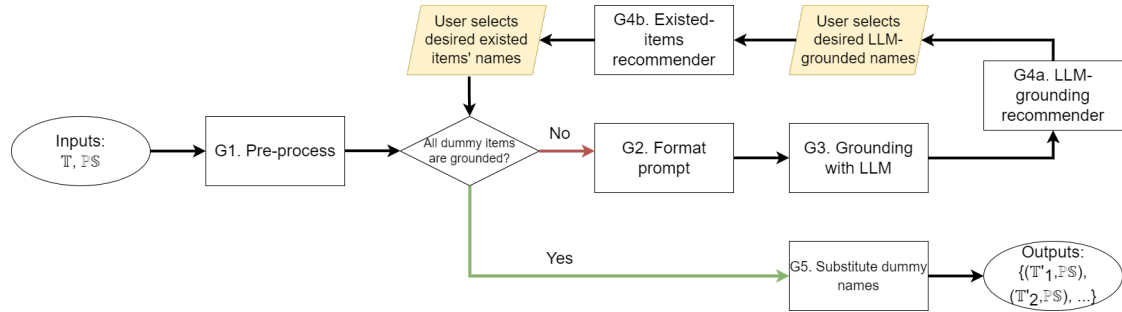


Figure 1: Flow chart of the grounding system. Modules involving the user’s inputs are coloured yellow.

Before grounding starts, ABC detects and repairs the fault in the input theory \mathbb{T} when it conflicts with the given preferred structure $\mathbb{P}\mathbb{S}$, which represents users’ observations [3]. Once the repair is done, users need to manually copy a repaired theory to start the grounding process.

We design a heuristic to formulate a prompt in the ungrounded theory. We suspect that LLMs perform grounding more accurately if we provide sufficient knowledge in the prompt, so we contain assertions without dummy items and multiple axioms with the same dummy item in the same prompt. For each proposition with a maximum arity of 3 in axioms, we convert it into natural language with the interpretation in Table 2 in the Appendix. The interpretation is similar to [14], except we substitute the dummy name by the item’s type - “property” for the dummy predicate, and “entity”/“kind” for the dummy constant. We gather the propositions into rules using conditional sentences and append the specification of the word limit at the end of the prompt to avoid LLMs returning lengthy answers. Example 2 illustrates the resulting prompt for grounding *dummyConst1* using the above heuristics, with setting the word limit as 5.

For each grounding of dummy items, users are presented with one to two rounds of recommendations. The initial recommendations are from the “LLM-grounding Recommender”, a phase where the LLM’s suggestions are displayed. Users are provided with the option to directly select the LLM-suggested answers, retain the dummy name, or propose new names. The inclusion of the latter choice allows users to refine the grounding names based on the LLM-generated answers or to tailor them to their preferences. In the subsequent phase “Existed-items Recommender”, the system explores the presence of existing items within the theory that exhibit high similarity to the chosen or suggested groundings from the previous phase. This comparison process involves assessing the resemblance of all prior suggestions against constants or predicates within the theory, depending on the type of dummy item. This phase employs the F1 score of BERTScore vanilla (referred to as F1 BERTScore) [15] to gauge the similarity between items. BERTScore utilizes embeddings from the pre-trained BERT model,

calculating the cosine similarity of embeddings to measure word matches between candidates and references [15]. Higher scores correspond to more significant similarity. Users also have the flexibility to set a threshold for the F1 BERTScore, enabling the system to recommend items that surpass the specified F1 BERTScore.

Example 2: Prompt Formulating in Repaired Tweety Theory

$\implies \text{bird}(\text{polly}, \text{dummyConst1})$
 $\text{bird}(X, Y) \implies \text{feather}(X)$
 $\text{bird}(X, \text{dummyConst1}) \implies \text{fly}(X)$
 $\implies \text{penguin}(\text{tweety})$
 $\text{penguin}(X) \implies \text{bird}(X, \text{flightless})$

Prompt: Given that tweety is a penguin. What is a possible **entity**, such that polly is a bird of **the entity**, and In a FOL expression, if X is a bird of **the entity**, then X can fly? Answer within 5 words^a.

^aNotice that the names of the penguins Tweety and Polly are not capitalized in the prompt.

An important consideration is that the grounding process has the potential to reintroduce faults into repaired theories. In response, we incorporate a safeguard as an extra feature by aiding users in re-running ABC following the exportation of all grounded theories. This validation step assesses whether the theories remain free from faults. This iterative approach involving repair and grounding persists until users attain a satisfactory theory. A practical example of the interplay between repair and grounding is depicted in the appendix.

3. Grounding Performance

We experimented with GPT-3.5 Turbo (4K context version) and GPT-4 (8K context version) [10, 16] using OpenAI’s ChatComplete API without further fine-tuning. As ABC is a domain-independent repair system, we intentionally omitted both fine-tuning and few-shot learning to gauge how these models perform without such adjustments.

As ABC uses Datalog, a subset of FOL, the grounding system cannot be evaluated with major FOL datasets [17, 18]. We compromised to examine the performance of LLMs. We substituted some items with dummy names in the theories and studied if the LLMs could ground similar items as the original ones in our system. We constructed theories automatically from two knowledge bases, enriched WebNLG dataset [19] and excerpt of DART [20], and replaced some items with dummy names. These theories serve as simulations of the generated repaired theories, with both assertions and rules. Details of the construction of the evaluation dataset are in the project’s GitHub repository¹.

We adopted two semantic-based metrics, F1 BERTScore [15] and SAS [21], to evaluate the semantic similarity of answers in LLM-grounding Recommender, as they are shown to have

¹<https://github.com/HistoChan/ABCgrounding>

a certain correlation between human judgement [21]. The former one is the same as the one used in the “Existed-items Recommender”. SAS, however, uses a pre-trained cross-encoder and applies the model by concatenating two texts with a separator token in between. Different from BERTScore, SAS considers two inputs together [21]. We calculated the scores of the LLM outcomes with the original items’ names. All the metrics values range from 0 to 1, with values closer to 1 indicating greater semantic similarity between candidates and references.

Prompt content	number of output	gpt-3.5-turbo		gpt-4	
		BERTScore	SAS	BERTScore	SAS
basic	1	81.78	8.63	81.73	11.35
w/ background	1	82.46	12.82	84.48	26.82
w/ multi axioms	1	81.89	9.54	82.27	15.41
w/ both	1	82.54	13.22	84.99	30.29
w/ both	3	84.51	20.31	86.47	37.06

Table 1

Performance of GPT models in LLM-grounding Recommender, where the metrics are in the micro average percentage point.

Table 1 is the statistics of the experiment, which shows that an LLM without any fine-tuning can still have an adequate grounding performance in zero-shot. The increase in the metrics confirms that the inclusion of additional background information in the prompt can obtain higher-quality grounding outcomes. The utilization of background information emerges as a more impactful hint for successful grounding, surpassing the effectiveness of querying multiple axioms in a single prompt. Moreover, the performances generally enlarge with model size, and an increase in the number of groundings would correspondingly enhance overall performance. We also adopted other models such as T5 models by Google [22], OpenLLaMA models from OpenLM Research [23], and Dolly 2.0 models by Databricks [24]. Their performances also support the above statements, which the project’s GitHub repository¹ contains the statistics of their performances. Some case studies are in the appendix.

4. Conclusions

In this paper, we have proposed a system of symbol grounding for the ABC repair system. We formulate the grounding challenge into a CBQA task and require LLMs to return possible answers. The system also embraces user interactivity, in which users have a right to control the model use, types of formatting prompts and grounding results. This system not only helps to enhance the semantics in the repaired theory but also determines the rationality of the repair plan. Yet, we suspect that the grounding performance is limited by the quality of the prompt, which can be improved in the future. Additionally, we facilitate using LLMs without either fine-tuning or few-shot learning for CBQA tasks by providing sufficient background information and enhancing the number of outputs. Nonetheless, we do not deem that few-shot learning can be replaced by providing background information. It is worth studying if few-shot learning can achieve similar performance of fine-tuning, and if the performance enhancement with few-shot learning is limited by the scale of the model.

References

- [1] A. Barr, E. A. Feigenbaum, *The handbook of artificial intelligence*, volume 1, Butterworth-Heinemann, 1981. URL: <https://www.sciencedirect.com/science/article/pii/B9780865760899500089>. doi:<https://doi.org/10.1016/B978-0-86576-089-9.50008-9>.
- [2] A. Bundy, X. Li, Representational change is integral to reasoning, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 381 (2023) 20220052. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2022.0052>. doi:10.1098/rsta.2022.0052.
- [3] X. Li, *Automating the Repair of Faulty Logical Theories*, 2021.
- [4] C. Sakama, K. Inoue, An abductive framework for computing knowledge base updates, *Theory and Practice of Logic Programming* 3 (2003). doi:10.1017/S1471068403001716.
- [5] S. O. Hansson, Ten philosophical problems in belief revision, *Journal of Logic and Computation* 13 (2003). doi:10.1093/logcom/13.1.37.
- [6] A. Bundy, B. Mitrovic, *Reformation: A Domain-Independent Algorithm for Theory Repair*, 2016.
- [7] S. Harnad, *The Symbol Grounding Problem*, 1990. URL: <http://cogprints.org/3106/>.
- [8] Q. Lyu, S. Havaladar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, C. Callison-Burch, Faithful Chain-of-Thought Reasoning, arXiv preprint arXiv:2301.13379 (2023). URL: <http://arxiv.org/abs/2301.13379>.
- [9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, *Advances in Neural Information Processing Systems* 35 (2022) 24824–24837. URL: <http://arxiv.org/abs/2201.11903>.
- [10] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, *Advances in neural information processing systems* 33 (2020) 1877–1901. URL: <http://arxiv.org/abs/2005.14165>.
- [11] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language Models as Knowledge Bases?, arXiv preprint arXiv:1909.01066 (2019). URL: <https://github.com/pytorch/fairseq>.
- [12] A. Roberts, C. Raffel, N. Shazeer, How Much Knowledge Can You Pack Into the Parameters of a Language Model?, *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference* (2020) 5418–5426. URL: <https://arxiv.org/abs/2002.08910v4>. doi:10.18653/v1/2020.emnlp-main.437.
- [13] S. Ceri, G. Gottlob, L. Tanca, What you always wanted to know about Datalog (and never dared to ask), *IEEE Transactions on Knowledge and Data Engineering* 1 (1989) 146–166. doi:10.1109/69.43410.
- [14] A. Mpagouli, Converting First Order Logic into Natural Language: A First Level Approach, in: *Current Trends in Informatics: 11th Panhellenic Conference on Informatics, PCI, 2007*, pp. 517–526.

- [15] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating Text Generation with BERT, arXiv preprint arXiv:1904.09675 (2019). URL: <https://github.com/Tiiiiger/bert>.
- [16] OpenAI, GPT-4 Technical Report, arXiv preprint arXiv:2303.08774 (2023). URL: <http://arxiv.org/abs/2303.08774>.
- [17] S. Han, H. Schoelkopf, Y. Zhao, Z. Qi, M. Riddell, L. Benson, L. Sun, E. Zubova, Y. Qiao, M. Burtell, D. Peng, J. Fan, Y. Liu, B. Wong, M. Sailor, A. Ni, L. Nan, J. Kasai, T. Yu, R. Zhang, S. Joty, A. R. Fabbri, W. Kryscinski, X. V. Lin, C. Xiong, D. Radev, FOLIO: Natural Language Reasoning with First-Order Logic, arXiv preprint arXiv:2209.00840 (2022). URL: <http://arxiv.org/abs/2209.00840>.
- [18] J. Tian, Y. Li, W. Chen, L. Xiao, H. He, Y. Jin, Diagnosing the First-Order Logical Reasoning Ability Through LogicNLI, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (2021) 3738–3747.
- [19] T. C. Ferreira, D. Moussallem, S. Wubben, E. Kraemer, Enriching the WebNLG corpus, in: Proceedings of the 11th International Conference on Natural Language Generation, 2018, pp. 171–176. URL: http://data.statmt.org/wmt17_systems.
- [20] L. Nan, D. Radev, R. Zhang, A. Rau, A. Sivaprasad, C. Hsieh, X. Tang, A. Vyas, N. Verma, P. Krishna, Y. Liu, N. Iwanto, J. Pan, F. Rahman, A. Zaidi, M. Mutuma, Y. Tarabar, A. Gupta, T. Yu, Y. C. Tan, X. V. Lin, C. Xiong, R. Socher, N. F. Rajani, DART: Open-Domain Structured Data Record to Text Generation, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 432–447. URL: <https://aclanthology.org/2021.naacl-main.37>. doi:10.18653/v1/2021.naacl-main.37.
- [21] J. Risch, T. Möller, J. Gutsch, M. Pietsch, Semantic Answer Similarity for Evaluating Question Answering Models, arXiv preprint arXiv:2108.06130 (2021). URL: <http://arxiv.org/abs/2108.06130>.
- [22] 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, Journal of Machine Learning Research 21 (2020) 1–67. URL: <http://jmlr.org/papers/v21/20-074.html>.
- [23] X. Geng, H. Liu, OpenLLaMA: An Open Reproduction of LLaMA, 2023. URL: https://github.com/openlm-research/open_llama.
- [24] M. Conover, M. Hayes, A. Mathur, X. Meng, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, R. Xin, Free Dolly: Introducing the World’s First Truly Open Instruction-Tuned LLM, 2023. URL: <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>.

A. Interpretation of a proposition in natural language

Proposition	Interpretation Based on the Part of Speech of Predicate's First Word	Examples
pred(Const)	Verb: Const pred . Others: Const is pred .	fly(bird) → “bird fly.”
pred(Const1, Const2)	Verb: Const1 pred Const2 . Others: Const is pred of Const2 .	capitalOf(london, england) → “london is capital of of england.”
pred(Const1, Const2, Const3)	Verb: Const1 pred (Const3) Const2 . Others: Const is pred (Const3) of Const2 .	mother(diana, william, dummyNormal) → “diana is mother (kind) of william.”

Table 2

Interpretations of propositions in natural language, where constants can be replaced by variables. Dummy items are renamed with their types, and grammar mistakes are ignored.

B. Example of Interplay of Repair and Grounding

We provide the highlight on how the repair and grounding processes collectively address conflicts and lead to the attainment of a desirable theory in Example 3.

Example 3: Repair and Grounding a Capital Theory

$$\begin{aligned}
 & \text{capitalOf}(X, Y) \wedge \text{capitalOf}(Z, Y) \implies X = Z \\
 & \implies \text{capitalOf}(\text{edinburgh}, \text{england}) \\
 & \implies \text{capitalOf}(\text{glasgow}, \text{scotland}) \\
 & \implies \text{capitalOf}(\text{london}, \text{england})
 \end{aligned}$$

$\mathcal{T}(\text{PS}) = \emptyset,$
 $\mathcal{F}(\text{PS}) = \{\text{edinburgh} = \text{london}, \text{london} = \text{edinburgh}, \text{glasgow} = \text{edinburgh}, \text{glasgow} = \text{london}, \text{edinburgh} = \text{glasgow}, \text{london} = \text{glasgow}\}$

Step 1: ABC finds there is a fault in having two capitals in England, which it suggests replacing “england” with “dummyEngland1” in $\text{capitalOf}(\text{edinburgh}, \text{england})$, and it is grounded as “scotland”:

$$\begin{aligned}
 & \text{capitalOf}(X, Y) \wedge \text{capitalOf}(Z, Y) \implies X = Z \\
 & \implies \text{capitalOf}(\text{edinburgh}, \text{scotland}) \\
 & \implies \text{capitalOf}(\text{glasgow}, \text{scotland}) \\
 & \implies \text{capitalOf}(\text{london}, \text{england})
 \end{aligned}$$

Example 3 (Continue): Repair and Grounding a Capital Theory

Step 2: ABC finds there is a fault in having two capitals in Scotland, which it suggests replacing “capitalOf” with “dummyPred” in *capitalOf(glasgow, scotland)*, and it is grounded as ‘cityOf’:
 $capitalOf(X, Y) \wedge capitalOf(Z, Y) \implies X = Z$

$\implies capitalOf(edinburgh, scotland)$

$\implies cityOf(glasgow, scotland)$

$\implies capitalOf(london, england)$

C. Examples of Grounding Results

We compare the performance of grounding containing background information of two models in Example 4, which illustrates that the grounding would be more reasonable with providing background information.

Example 4: A comparison of grounding answers of Example 2

- Without extra content: What is a possible entity, such that opus is broken wing of the entity? Answer within 5 words^a.
 - GPT-3.5 Turbo: Defective wing.
 - T5 Large (NQ): Feathers are
- With background information: Given that opus is super penguin. What is a possible entity, such that opus is broken wing of the entity? Answer within 5 words^a.
 - GPT-3.5 Turbo: X is injured
 - T5 Large (NQ): Cannot fly

^aNotice that the name of the penguin Opus is not capitalized in the prompt.

We provide Example 5 as a comparison of grounding performance on different LLMs, in which the answers are more accurate with larger model sizes. Despite our explicit instruction to return answers within five words and request of the maximum output tokens as five, Dolly 2.0 and OpenLLaMA still have a high tendency to return a complete sentence.

Example 5: A comparison of grounding answers of a repaired Capital Theory

Question: What is a possible entity such that edinburgh is cap of of the entity? Answer within 5 words.

- Dolly 2.0 3B: Edinburgh is the capital of
- Dolly 2.0 7B: The answer is the Edinburgh
- OpenLLaMA 3B: edinburgh is cap of
- OpenLLaMA 7B: The answer is Scotland.
- GPT-3.5 Turbo & T5 XL (NQ): Scotland
- GPT-4: Scotland or United Kingdom.
- T5 Small & Large : Edinburgh
- T5 Small (NQ): other social entity
- T5 Large (NQ): Kingdom of Scotland

We experimented on the effect of the number of groundings generated from LLM. This experiment lay in the diversity of the returned results. With open-ended questions like that in Example 6, the answers returned reflect the multifaceted nature of potential responses. The augmented number of returned answers not only aids in identifying high-quality grounding but also empowers users to brainstorm a broader spectrum of possible groundings.

Example 6: A comparison of answers of prompt from a repaired Tweety Theory

Question: Given that tweety is penguin. What is a possible entity such that polly is bird of the entity, and In a FOL expression, if x is bird of the entity, then x is fly? Answer within 5 words.^a.

- GPT-3.5 Turbo: “Airplane.”, “Flying creature like parrot”, “Fish”
- GPT-4: “Sky or Air could be”, “Possible entity is 'the’”, “The possible entity: magical”
- Suggested Answer: “flying”

^aNotice that the names of the penguins Tweety and Polly are not capitalized in the prompt.