Can LMs Store and Retrieve 1-to-N Relational Knowledge?

Haruki Nagasawa[1] **Benjamin Heinzerling**[2,1] **Kazuma Kokuta**[1] **Kentaro Inui**[1,2]
[1]Tohoku University [2]RIKEN
{haruki.nagasawa.s8, kokuta.kazuma.r3}@dc.tohoku.ac.jp
benjamin.heinzerling@riken.jp kentaro.inui@tohoku.ac.jp

Abstract

It has been suggested that pretrained language models can be viewed as knowledge bases. One of the prerequisites for using language models as knowledge bases is how accurately they can store and retrieve world knowledge. It is already revealed that language models can store much 1-to-1 relational knowledge, such as "country and its capital," with high memorization accuracy. On the other hand, world knowledge includes not only 1-to-1 but also 1-to-N relational knowledge, such as "parent and children." However, it is not clear how accurately language models can handle 1-to-N relational knowledge. To investigate language models' abilities toward 1-to-N relational knowledge, we start by designing the problem settings. Specifically, we organize the character of 1-to-N relational knowledge and define two essential skills: (i) memorizing multiple objects individually and (ii) retrieving multiple stored objects without excesses or deficiencies at once. We inspect LMs' ability to handle 1-to-N relational knowledge on the controlled synthesized data. As a result, we report that it is possible to memorize multiple objects with high accuracy, but generalizing the retrieval ability (expressly, enumeration) is challenging.

1 Introduction

As a result of their pretraining on large amounts of text, language models (LMs) store certain world knowledge facts, such as "Paris is the capital of France", in their parameters and can retrieve that knowledge when given a suitable prompt. Since the ability to store and retrieve knowledge is also a key functionality of knowledge bases (KBs; Weikum et al., 2021), prior work has proposed to view language models as knowledge bases (Petroni et al., 2019). Quantitative evaluation of world knowledge in LMs has focused on 1-to-1 relational knowledge involving two entities, such as a country and its capital (Petroni et al., 2019; Heinzerling and Inui, 2021; Safavi and Koutra, 2021; Razniewski et al., 2021). However, the question if and how well LMs can handle 1-to-N relations, such as relations between parents and their children, is underexplored so far.

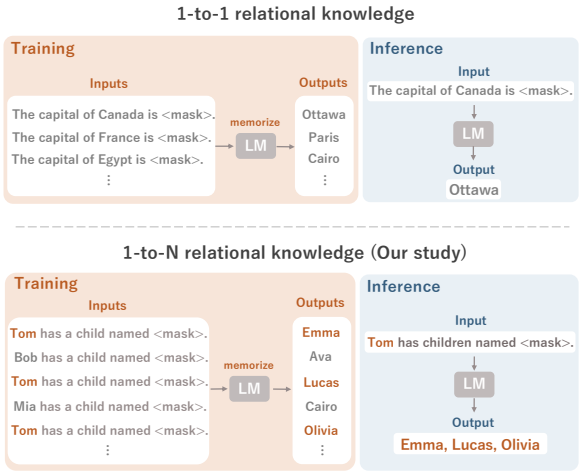


Figure 1: Memorize and enumerate relational knowledge. We are considering a synthetic setting in which the LM is made to memorize a specific set of individual relations and then needs to aggregate those relations into 1-to-N relations.

Here, we conduct a study to assess the capability of LMs to store and retrieve 1-to-N relations in a manner similar to knowledge bases. We consider a setting in which the model first is trained to memorize individual relation instances, such as "Tom has a child named Emma", "Bob has a child named Ava", "Tom has a child named Lucas", and "Tom has a child named Olivia". During inference the model then has to retrieve 1-to-N relation, e.g., "Tom has children named Emma, Lucas, Olivia" (Figure 1).

To investigate the possibility of viewing LMs as KBs more precisely, it is necessary to clarify the basic abilities of LMs, such as how accurately they can store 1-to-N relational knowledge and how flexibly they can retrieve multiple entities they have stored.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics - Student Research Workshop, pages 130–138

Our study represents the first comprehensive investigation of 1-to-N relational knowledge. Our contributions are summarized as follows: (1) We identified the capabilities necessary for LMs to handle 1-to-N relational knowledge, taking into account its unique properties. Specifically, LMs must be able to accurately memorize any object appearing discretely and enumerate multiple objects without over- or under-recall based on memory. (§ 3) (2) Based on the identified capabilities, we formulated two training schemes: element-valued supervision for "memorization" and set-valued supervision for "enumerating." (§ 4) (3) We conducted a quantitative evaluation of LMs' "memorization" abilities from both subject-oriented and object-oriented perspectives and categorized the errors encountered during "enumerating." Our results suggest that LMs are able to store 1-to-N relational knowledge with reasonable accuracy, but generalizing the ability to enumerate proves to be challenging. (§ 6)

2 Related Work

Factual knowledge probing Petroni et al. (2019) investigated how much knowledge LMs had acquired from large corpora by having models such as pretrained BERT (Devlin et al., 2019) solve problems in the "fill-in-the-blank" format. They also pointed out three critical advantages of treating LMs as KBs: "LMs require no schema engineering, do not need human annotations, and support an open set of queries."

Jiang et al. (2020) and Brown et al. (2020) also worked on creating optimal prompts for extracting correct answers from pretrained LMs. These investigations aim to extract knowledge that LMs have acquired implicitly during pretraining. On the other hand, we are interested in the degree to which knowledge can be handled accurately when LMs explicitly learn it. Thus, investigating what and how well pretrained LMs acquire 1-to-N relational knowledge from corpora is beyond our scope.

Storing 1-to-1 relational knowledge Heinzerling and Inui (2021) established two basic requirements for treating LMs as KBs: "(i) the ability to store a lot of facts involving a large number of entities and (ii) the ability to query stored facts." Based on these requirements, they elaborately examined how much and how accurately LMs can store 1-to-1 relational knowledge by comparing various entity representations. However, the behavior of LMs concerning 1-to-N relational knowledge remains unclear.

Set handling This study explores handling multiple objects, which can be achieved by handling a set of objects. Previous works such as Deep Sets (Zaheer et al., 2017) and Set Transformer (Lee et al., 2019) are representative ones that address set handling in neural networks or transformers (Vaswani et al., 2017).

Both focus on sets as inputs, being permutation-invariant and treating sets of arbitrary size. While this study focuses on sets as outputs rather than inputs, the properties such as permutation-invariant are considered to be essential aspects in common.

3 Designing an approach to 1-to-N relational knowledge

In this section, we describe the unique properties of 1-to-N relational knowledge and what capabilities of LMs are needed to handle 1-to-N relational knowledge.

To begin with, we define three significant unique factors that make 1-to-N relational knowledge challenging to deal with: First, when the subject or relation under consideration changes, the number of objects associated with it changes. For example, consider answering the question, "{Subject} has children named <mask>." The difficulty is that the number of correct objects changes depending on the input. Second, considering existing corpora, multiple objects are likely to occur discretely. For example, Barack Obama has two children, Malia and Sasha, but only Malia may appear in some specific contexts, and only Sasha may appear in other contexts.. Finally, third, when we assume a situation where an LM is used practically as a KB, it is necessary to output these discretely appearing objects together to avoid generating an inadequate response to the input query.

Therefore, given the above properties, the two essential LMs' competencies considered necessary to manage 1-to-N relational knowledge are as follows. (i) "the ability to accurately memorize any objects appearing discretely." (ii) "the ability to retrieve multiple objects without over- or under-recall based on memory." In order to consider an end-to-end approach to 1-to-N relational knowledge, this study tackles it as a generative task using the sequence-to-sequence model (Sutskever et al., 2014), which allows for flexible responses based on input.

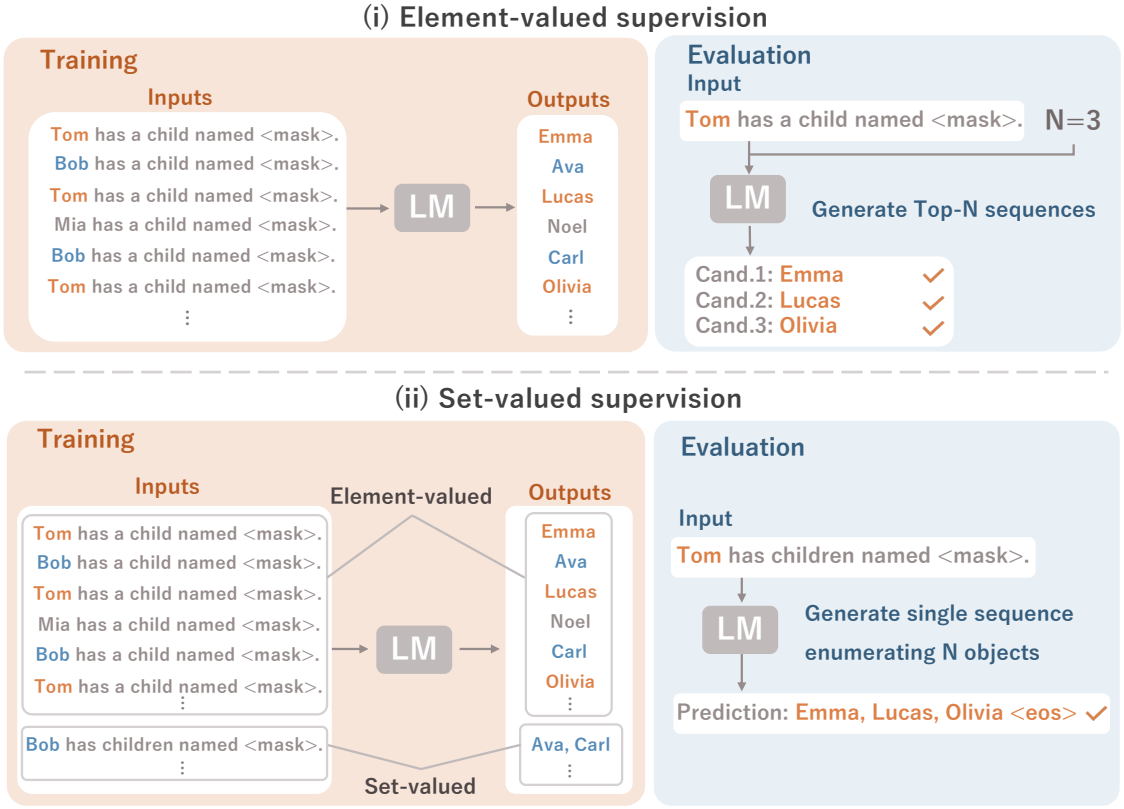


Figure 2: (i) Element-valued supervision and (ii) set-valued supervision. **Element-valued supervision** is intended to have the LM memorize all objects of a 1-to-N relation individually. For a given subject, there are as N relation instances. We train the model to output a single object entity when given an input query about a subject entity. During the evaluation, N sequences are generated using a beam search of size N to verify if all N object entities are stored and retrieved. **Set-valued supervision** is used to train the model to enumerate all objects for a given entity and predicate in one prediction step.

4 Method

4.1 Terminology

In this work, we make use of the following terms:

Relation triple: A triple consisting of a *subject* and an *object* entity, as well as a predicate that describes the relation that holds between the subject and the object, e.g., (Tom, hasChild, Emma).

1-to-N relation: A set of relation triples with the same subject and predicate, but different objects, e.g., (Tom, hasChild, Emma) and (Tom, hasChild, Lucas).

Individual relation instance: A relation triple expressed in text, for example "Tom has a child named Emma."

Element: Viewing a 1-to-N relation as a set, we refer to individual relation instances as *elements* of that set, e.g., "Tom has a child named Emma." is an element of the 1-to-N relation that holds between Tom and his children.

Element-valued supervision: One of the two supervised training schemes we employ. A model is trained on elements, i.e., individual relation instances, of 1-to-N relations. Concretely, the model is given a relation instance with the object masked out, e.g., "Tom has a child named <mask>." and has to predict the masked out object, e.g., "Emma". The goal of this training scheme is to have the model memorize individual objects based on their corresponding subjects.

Set-valued supervision: In the second of our supervised training schemes the model is trained to predict the set of all objects for a given subject and predicate, e.g., given "Tom has children named <mask>.", the model has to generate the text "Emma, Lucas, Olivia".

Table 1: Templates: We used different templates for each model to fit each pretraining setting.

		Parent-children	Director-titles
BART	Element-valued supervision	{Sbj} has a child named <mask>.	{Sbj} directed a film titled <mask>.
	Set-valued supervision	{Sbj} has children named <mask>.	{Sbj} directed following movies: <mask>.
T5	Element-valued supervision	What is the name of {Sbj}'s child?	What movie did {Sbj} direct?
	Set-valued supervision	What are the names of {Sbj}'s children?	What are the titles of movies {Sbj} directed?

4.2 Handling of 1-to-N Relational Knowledge

We investigate the behavior of LMs for 1-to-N relational knowledge when explicitly trained. Specifically, we use the sequence-to-sequence model to generate variable-length responses to inputs.

As described in § 3, the two abilities necessary for LMs to handle 1-to-N relational knowledge are (i)memorizing multiple discretely appearing objects and (ii)enumerating memorized objects without excess or deficiency. In this section, we conduct two experiments, each corresponding to the essential abilities.

(i) Memorization The first experiment is aimed at "memorization" through element-valued supervision. Here, 1-to-N relational knowledge is decomposed into a one-to-one form, and we train LMs to memorize multiple objects individually. In the learning process, one object is output in response to an input for a particular subject, and then all objects will be memorized in this fashion. Therefore, the state in which the LMs memorize all N objects can also be paraphrased as the state in which the LMs can output all N objects.

Therefore, the evaluation of whether LMs memorized multiple objects is checked by generating multiple sequences using beam-search. Specifically, N sequences are generated for a subject using the same query as the training data. By checking how many correct objects are included in the sequences, we evaluate how many objects the LMs memorized.

(ii) Enumeration The second experiment attempts to acquire "the ability to enumerate memorized objects." Here, training by set-valued supervision is performed in conjunction with memorization by element-valued supervision. The reason for using the two supervisory methods together is the premise that to enumerate multiple objects, it is necessary to memorize them in the first place. Although it is possible to perform element-valued supervision and then shift to set-valued supervision, catastrophic forgetting of memorized objects may occur during the training of set-valued supervision. Indeed, we have confirmed that catastrophic forgetting of memorized objects occurs during set-valued supervision, so in this paper, the two supervisory methods are used together. For some subjects in the training data, LMs explicitly learn the behavior of enumerating the objects in response to queries that explicitly ask for multiple objects. We then test whether set-valued supervision allows LMs to enumerate objects for other subjects as well, i.e., whether they can generalize the ability to enumerate.

5 Experimental setup

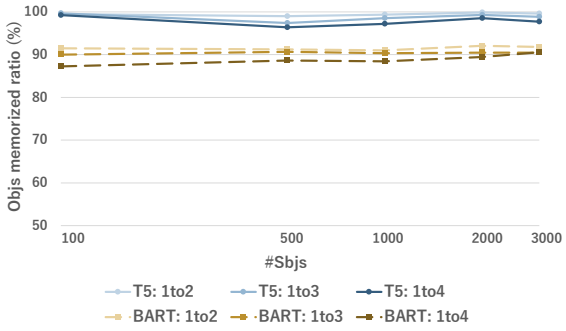
5.1 Synthetic Data

In the following experiments, we uniquely prepared the 1-to-N dataset to measure how well LMs can accurately store plenty of facts. Specifically, we randomly obtained canonical names of parents and their two to four children from Wikidata (Vrandečić and Krötzsch, 2014). We also randomly obtained the canonical names of directors and their two to four representative films from IMDb Datasets[1]. Therefore, by preparing 1-to-2, 1-to-3, and 1-to-4 relational knowledge, we will observe how LMs performance changes as the number of objects increases. We only collected data that meets the following conditions.

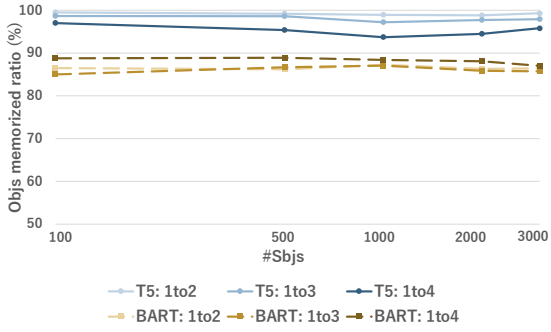
- To ensure that all entities are distinguishable, there is no data with the same canonical name across both subjects and objects.
- Only entities consisting of four or fewer words separated by spaces or hyphens are used to adjust for storing difficulty due to word length.

We only consider memorizing and enumerating entities which appear in the training data.

[1]https://www.imdb.com/interfaces/

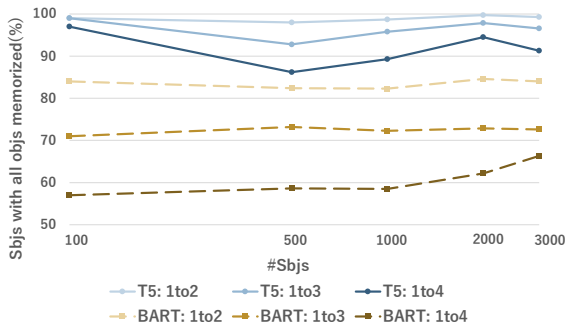


(a) Parent-children dataset

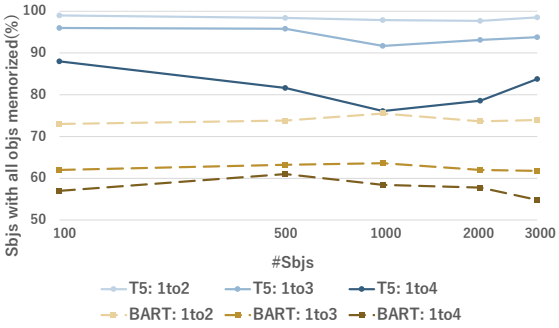


(b) Director-titles dataset

Figure 3: Object-oriented memorization accuracy: showing how many objects LMs memorized



(a) Parent-children dataset



(b) Director-titles dataset

Figure 4: Subjects-oriented memorization accuracy: showing how many subjects are there that LMs memorized their corresponding N objects.

5.2 Models and Training settings

We used the pretrained BART-base (Lewis et al., 2020) and T5-base (Raffel et al., 2019) as the sequence-to-sequence model in the experiments. The training in the two experiments described below (§ 6.1 and § 6.2) was continued until the models strongly overfit the training data. Precisely, we continued training until the accuracy of the training data no longer improved by more than 30 epochs.

The accuracy was calculated as follows: for element-valued supervision, the accuracy was determined by whether the model could generate the correct object for each subject in the input. If the model generated one of the correct N objects for each subject, it was considered correct; otherwise, incorrect. For set-valued supervision, the accuracy was determined by whether the model generated a set of multiple correct objects with no omissions or additions. If the model generated a complete set of correct objects, it was considered correct; otherwise, incorrect.

As detailed training settings, the learning rate was started at 5e-5 in common with BART and T5, and it was reduced by half if the accuracy did not improve by more than three epochs. The batch size was varied according to the model and training data size/domain. AdamW (Loshchilov and Hutter, 2019) was commonly used as the optimizer. In addition, a different template was used for each model so that the input sentence templates were similar to the pretraining settings for each (BART uses <mask> token in pretraining, but T5 does not.) The templates used are listed in Table 1.

6 Experiments

6.1 Element-valued supervision

In the first experiment, we investigated the ability to memorize multiple objects using element-valued supervision. Here, we tested whether the LMs could correctly store N objects associated with a single subject. Specifically, as shown in Figure 2, the learning process of having one object generated for each input sentence, such as "{Subject} has a child named <mask>." or "{Subject} directed a film titled <mask>." was performed for all objects. Thus, the learning setup is such that there are as many target sentences as objects for each input sentence.

Table 2: Accuracy of enumerate operation

	Model	BART-base			T5-base		
	Set-valued supervision ratio	30%	60%	90%	30%	60%	90%
Parent-children	1-to-2	46.7	45.8	49.3	27.0	40.7	**49.5**
	1-to-3	8.33	9.33	9.67	10.7	16.8	**20.7**
	1-to-4	1.00	1.33	2.17	0.500	2.33	**2.67**
Director-titles	1-to-2	42.0	43.3	**44.17**	19.8	24.2	28.7
	1-to-3	22.5	24.2	**26.3**	14.8	15.8	23.7
	1-to-4	6.17	10.7	**11.3**	2.33	3.83	7.00

We then checked the degree to which LMs trained with element-valued supervision could recall multiple objects through the generation of N sequences using beam search. To be precise, N was for the number of objects associated with the input subject, and we analyzed the count of correct objects within those sequences.

In this experiment, we also tested whether the LMs' memorization accuracy changed when the training data size, i.e., the number of entities, was varied. Here, we evaluated this memorization accuracy from two perspectives.

Object-oriented memorization accuracy The first perspective is object-oriented memorization accuracy, shown in Figure 3, which evaluates the degree of recall of objects in the training data. Figure 3a and 3b correspond to the parent-children and director-titles datasets, respectively. The solid blue line corresponds to T5, and the dashed yellow line to BART, with darker colors corresponding to 1toN relational knowledge with more objects. The results show that T5 has better memorization accuracy than BART, although no significant differences by data domain were observed. Also, the larger N, i.e., the greater the number of objects associated with one subject, the more likely N entities could not be memorized.

Subject-oriented memorization accuracy The second perspective, subject-oriented memorization accuracy, evaluated how many subjects were memorized with all related N objects. Specifically, in generating multiple objects by beam search, we show how many subjects existed for which all N objects were generated.

The results are shown in Figure 4, where 4a and 4b correspond to the parent-children and director-title datasets, respectively, as in Figure 3. The results confirmed that, overall, T5 has higher memorization accuracy. Looking at performance by the number of objects, it is clear that, in common with the two data domains and two models, the greater the number of objects, the more difficult it was to remember all of them in conjunction with the subject.

Interestingly, both memorization accuracies in the two perspectives show roughly independent behavior concerning data size. One possible reason for the higher overall memory accuracy of T5 is that the parameter size of the T5-base is about 1.5 times larger than that of BART-base. This may contribute to higher memory accuracy. The fact that 100% memorization accuracy was not achieved for either data size may suggest that memorizing 1-to-N relational knowledge is not easy for LMs. Examples of LMs' predictions are shown in Table 3.

6.2 Element-valued and Set-valued supervision

In this subsequent experiment, the model was trained with element-valued and set-valued supervision to acquire the ability to enumerate all associated objects. More expressly, compared to the first experiment, we additionally employed set-valued supervision, which involved using "{Subject} has children named <mask>." as the input sentence and "{Object1}, {Object2}, ..." as the corresponding target sentence, as an example. This approach aimed to generalize the model's ability to enumerate all accurately memorized objects in response to queries requesting multiple objects.

We conducted both element-valued and set-valued supervision during training. Specifically, we trained LMs using element-valued supervision on all subjects to memorize all associated objects. We fixed the training data size at 3000 subjects for each. Simultaneously, we randomly selected 20% of the subjects, i.e, 600 subjects, as a test set for set-valued supervision. For the remaining 80% of

Table 3: Examples of generated N sequences for element-valued supervision. Showing 1-to-3 relational knowledge, which includes leakage of memorization. Objects with green background color are correct and those with red are incorrect.

Data Domain	1-to-N	Subject	Gold objects	Top-N sequences
Parent-children	1-to-3	Dr. Dre	Hood Surgeon La Tanya Danielle Young Truice Young	BART 1: Hood Surgeon 2: Truice Young 3: Young Hood Surgeon T5 1: Hood Surgeon 2: Truice Young 3: La Tanya Danielle Young
Director-titles	1-to-3	Jack Holton	A Dream for Christmas Escape to Witch Mountain The Wild Country	BART 1: Escape to Witch Mountain 2: A Dream for Christmas 3: The Wild Country T5 1: Escape to Witch Mountain 2: A Dream for Christmas 3: Adventures in Dinosaur City

the subjects, we varied the proportion of subjects for which set-valued supervision was applied (i.e., 30%, 60%, or 90%) to examine whether the generalization ability would change depending on the number of instances that the LMs learned how to enumerate their corresponding objects.

The goal was to investigate how well the model could generalize to subjects in the test set when using set-valued supervision and to determine the impact of varying the proportion of subjects with set-valued supervision on model performance.

The results (Table 2) show that the enumerating accuracy is highest when the supervision ratio is 90% for all, indicating that it is important to have many training instances to generalize the enumerating capability.

Although there are differences in the enumerating accuracy scores across data domains and models, we found a tendency for the enumeration performance to decrease significantly as the number of target words increases.

Error analysis Quantitative error distributions are shown in Table 4, and specific examples of incorrect answers are shown in Table 5. Table 4 shows that for small numbers of objects (e.g., 1-to-2), BART tended to generate incorrect objects (labeled "Incorrect"), while T5 often duplicated the same object (labeled "Duplication"), highlighting a noticeable difference between the two models. As the number of objects increased (e.g., 1-to-3, 1-to-4), both models were more likely to produce wrong answers due to missing objects (labeled "Missing"). The distribution of errors across different datasets was generally similar, but both models were more prone to missing objects in the parent-children dataset, suggesting that the type of entity names might have an impact on the error patterns.

7 Conclusion

We addressed handling 1-to-N relational knowledge by a generative approach using the sequence-to-sequence model. Since little work has been done on 1-to-N relational knowledge in previous studies, we started by organizing the properties of 1-to-N relational knowledge and setting up the capabilities considered necessary for LMs based on these properties.

Specifically, we defined two essential capabilities: "memory of discretely appearing multiple objects" and "enumeration of objects based on memory." Then, we developed training schemes based on these perspectives. We used element-valued supervision and beam search for the former to memorize and evaluate multiple objects. We found that nearly 90% of the objects could be memorized, although we observed a tendency for memory omissions to occur as the number of objects increased. However, we also confirmed that it is challenging to achieve 100% perfect memory.

For the latter, we attempted to generalize "enu-

Table 4: Quantitative error analysis on 90% set-valued supervision: showing the number of incorrect responses generated by the model, categorized into three types of errors. "Incorrect" denotes model-generated sequences that contain one or more incorrect objects. Responses that lack objects are classified as "Missing" (omission of objects), while those with duplicate instances of the same object are labeled as "Duplication."

	Model	BART-base			T5-base		
	Error Type	Incorrect	Missing	Duplication	Incorrect	Missing	Duplication
Parent-children	1-to-2	280	0	18	154	2	147
	1-to-3	229	306	7	93	287	96
	1-to-4	175	406	6	105	380	99
Director-titles	1-to-2	298	0	37	156	1	271
	1-to-3	70	352	20	41	287	130
	1-to-4	25	481	25	37	441	80

Table 5: Examples of enumerating error for the parent-children dataset. The error part is colored in red. These errors are for 1-to-3 relational knowledge and were generated by the T5, which is trained with 90% set-valued supervision.

Error	Subject	Gold and Prediction
Missing	Jeb Bush	Gold: George P. Bush, Noelle Bush, John Bush Jr. Pred: John Bush Jr., Noelle Bush (missing)
Incorrect	Shimon Peres	Gold: Tsvia Walden, Hemi Peres, Yoni Peres Pred: Tsvia Walden, Yoni Peres, Leo Peres
Duplication	Alice Meynell	Gold: Viola Meynell, Everard Meynell, Madeline Lucas Pred: Viola Meynell , Madeline Lucas, Viola Meynell
Excess(Incorrect)	Alan Alda	Gold: Beatrice Alda, Elizabeth Alda, Eve Alda Pred: Elizabeth Alda, Beatrice Alda, Eve Alda, Nanna Alda

meration ability" by set-valued supervision in conjunction with memorization by element-valued supervision. The results showed that learning more data improved the generalization performance for acquiring enumeration ability. However, we also observed the LM's behavior, which aligns with human intuition: the more objects increase, the more difficult it becomes to enumerate all of them correctly. Notably, the generalization performance for 1-to-2 relational knowledge was only about 50% for the test set, and for 1-to-4 relational knowledge, only about 10% generalization performance at most.

For our next steps, we are considering the following approach. The training setup of the current element-valued supervision is characterized by multiple target sentences for one input sentence, which is incompatible with the model's learning algorithm. Therefore, we would like to test a memorizing method using ordinal numerals such as first and second to distinguish each template for N objects. We would also like to investigate this memorization method's effect on the generalization performance of enumeration.

As for enumeration, which has been difficult to generalize, we would like to examine effective means of improving performance for a small number of objects. Specifically, we are considering adjusting the hyperparameters for text generation and verifying whether errors in enumerating will be reduced. After that, we would like to explore learning methods to enumerate N objects without needing hyperparameters adjustment in stages.

Introducing our 1-to-N problem setting into the LMs-as-KBs paradigm opens up many more intriguing challenges. While we investigated this setting under a controlled condition with a uniform frequency of object appearance, the frequency of each of the N objects in a corpus is likely to vary in reality. Furthermore, there may be multiple paraphrases expressing the same relation.

For example, in our study, we only considered the phrase "{Subject} has a child named {Object}." but there are other phrases such as "{Subject}'s child is {Object}." or "{Object} is a daughter of {Subject}." As a primary avenue for future research, we will explore whether LMs can handle 1-to-N relational knowledge effectively under these more complex conditions.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 21K17814 and JST CREST Grant Number JPMJCR20D2, Japan.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1772–1791. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Simon Razniewski, Andrew Yates, Nora Kassner, and Gerhard Weikum. 2021. Language models as or for knowledge bases. *CoRR*, abs/2110.04888.

Tara Safavi and Danai Koutra. 2021. Relational world knowledge representation in contextual language models: A review. *CoRR*, abs/2104.05837.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Found. Trends Databases*, 10(2-4):108–490.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. 2017. Deep sets. *CoRR*, abs/1703.06114.