Examining 4-bit quantized Qwen3-8B for object prediction @ LM-KBC 2025

Irene Mary Sam[1,†]

[1]MosAIk, Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France

Abstract

This is the system description paper for the solution that was presented as part of the LM-KBC challenge at ISWC 2025. The challenge involved predicting the object(s) of a triple given the subject and the relation, and can thus be considered a knowledge base completion task. This paper contains details on the proposed solution, and the results of that system based on the evaluation metrics and test data provided by the organisers. The approach adopted was one that involved manual prompt design, and the results obtained include modest gains on both recall as well as the macro F1 score. Also reported are some additional experiments regarding quantization and prompting strategy.

1. Introduction

Considering a knowledge base as a large collection of disambiguated, linked data that are stored as triples, the 4th edition of the LM-KBC challenge consisted of object prediction given the subject and the relation of a triple [1]. In the example given below as obtained from the train set, given *Punathil Kunjabdulla* as subject and *personHasCityOfDeath* as relation, the task is to predict *Kozhikode* as the correct object to complete this triple.

<"Punathil Kunjabdulla", "personHasCityOfDeath", "Kozhikode">

In this edition, emphasis was placed on probing a given, fixed LLM common to all, so as to understand the knowledge stored within said LLM and evaluate its capabilities with respect to the object prediction or knowledge base completion task. The following sections describe the task and data, the approach adopted, the results and a discussion of the same.

2. Background

Knowledge base construction as an emergent property of LLMs began to gain traction with the work of [2] who studied the possibility of using language models as knowledge bases on the premise that, in using large textual corpora to pretrain language models, the models tend to store relational knowledge present in the corpora, thus displaying a strong ability to recall factual information when probed.

Further evolution of LLMs saw them being used to automatically construct knowledge bases like in the case of LLM2KB and GPTKB [3][4]. Challenges still remain as seen from previous works, with entity and relation recognition remaining a tough barrier, apart from usual issues with LLMs related to consistency and correctness.

In a bid to understand this better, the various editions of the LM-KBC challenge have addressed the task of knowledge base completion in various ways. In this edition specifically, Qwen3-8B was mandated as the base model from which all analyses and outputs should stem, with other restrictions including disallowing fine-tuning and retrieval-augmented methods [1]. The following section provides more details about the task at hand, as well as the method adopted.

Joint proceedings of KBC-LM and LM-KBC @ ISWC 2025

[†] The author's RA position is funded by the DGA-RAPID project SUMINO. Part of the experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

irene-mary.sam@loria.fr (I. M. Sam)

3. System description

3.1. Task

The task is a knowledge base completion / object prediction task, where, given the subject and relation of a triple, the goal is to predict the correct object(s) that fit the subject and relation. These can be single, multiple or *null* objects depending upon the subject and the relation, as seen below from the train set examples in Table 1.

Table 1
Some Examples of Relation Types

Relation	Subject	Object
hasCapacity	Changwon Stadium in Changwon	"27085"
companyTradesAtStockExchange	Erie Railroad	*null*
countryLandBordersCountry	Nicaragua	"Costa Rica", "Honduras"

These were the constraints that were specific to the challenge this year: (1) All participants had to use the Qwen3-8B model in order to complete the task, since the challenge focused on interrogating the internal knowledge of this specific model, which results in the performance of the various systems becoming easier to compare with respect to one another. (2) Fine-tuning and the usage of retrieval-augmented methods were disallowed, thus resulting in a situation where the model alone is involved in the prediction of objects. (3) Entity disambiguation of the predicted objects was not required as the evaluation would be done on the predicted object strings directly [1]. Further details on the data and the system are given below.

3.2. Data

The provided data consisted of subject-predicate-object triples, with each triple being a JSON object. It was split into train, validation and test sets with sizes as shown in Table 2, with each split containing various instances of the six relations as indicated in Table 3. The training and validation datasets carried the object as well alongside the subject and predicate, enabling an evaluation of the system at hand before running it on the test set.

Table 2
Dataset sizes

Dataset	Instance Counts
Train	478
Validation	478
Test	477
Total	1433

Table 3
Relation-specific instance counts by dataset type, alongside object datatypes

Relation	Train	Val	Test	Object Datatype
hasArea	100	100	100	numeric
hasCapacity	100	100	100	numeric
companyTradesAtStockExchange	100	100	100	string
personHasCityOfDeath	100	100	100	string
awardWonBy	10	10	10	string
countryLandBordersCountry	68	68	67	string

A peculiarity of the relations provided is that the objects to be predicted can be of different datatypes depending upon the relation, ranging from numeric to textual as seen in Table 3. Another defining feature is that *null* values and multiple objects per subject-relation pair are also possible in this dataset, thus increasing the difficulty of the knowledge base completion task.

3.3. System / Prompts

In approaching the object prediction task, the idea was to write prompts that perform better than the baseline. As the baseline prompts were quite curt and modelled along the lines of the fill-mask task reminiscent of [2], the prompts for this system were initally designed to be explicit and detailed in its ask for information as opposed to the baseline template, thus modelled similarly to the most explicit level of prompting in the work of [5]. While this produced satisfactory results, upon further examination of the main and thinking mode outputs of the model, it was found that the model struggles in the case of *awardWonBy* where it collapses into a thought loop more often than not, as well as in differentiating between single and multiple object relations. Below is a brief example of the outputs when the model was prompted with the relation *awardWonBy* and the subject *Fulbright Prize*:

> <think> Okay, the user is asking for the object of the triple where the subject is "Fulbright Prize" and the relation is "awardWonBy". So, I need to find out who has won the Fulbright Prize. ...
>
> The Fulbright Prize is typically awarded to individuals, so the objects would be the names of people who have received it. ...
>
> However, the user might be looking for specific examples. I know that notable recipients include people like the Dalai Lama, who received the Fulbright Prize in 1989. There's also the case of the author who won the prize but was later expelled from the U.S., though I might be mixing up details here. ...
>
> Alternatively, maybe the user is referring to the Fulbright Scholarship rather than the prize. The term "prize" might be a bit confusing here. The Fulbright Program is more of a scholarship, but sometimes it's referred to as a prize. ...
>
> In this case, since I can't provide a comprehensive list, but I know the Dalai Lama is one, I should include that. However, the user might expect more. But given the time constraint, I'll provide the Dalai Lama as the object. </think>
>
> The Dalai Lama, John F. Kennedy, Eleanor Roosevelt, and others.

Thus, the final system contains the following mix of strategies: instead of having a single system prompt, there are now three. *awardWonBy* is treated as a special case and given its own system prompt. Similarly, *countryLandBordersCountry* and *companyTradesAtStockExchange* are treated as cases with multiple objects per triple and given a system prompt accordingly, and the rest are given the default system prompt. In designing these system prompts, some personalisation was used in declaring the specialty of the system with respect to the data it handles, similar in strategy to that of the role-play prompting discussed in [6]. An example of one of these system prompts can be found in Appendix B.

In addition to these 3 system prompts, all six relations have their own user prompts as well[1], some examples of which can be found in Appendix C. No additional post-processing steps were employed; the output from the model, after removing the thinking mode part of the output, is kept as is.

3.4. Implementation

While a couple of other tweaks were considered, the details of which are in Section 6, the above returned better results, hence that was fixed as the main system. Prompt-tuning, optimisation methods and frameworks such as DSPy were not used in light of the instructions that prohibited fine-tuning, since,

[1] The prompt templates and code are available on GitHub

while they do not change the original model parameters, they still involve adding learnable parameters to the input embeddings in a bid for soft-prompting, thus fine-tuning extra-model parameters [7][8].

In running the Qwen3-8B model for prediction, it was noted that using the hyperparameters listed in its documentation was key to obtaining good results, as, without it, the system tended to either take too long to think or collapse into incomprehensible output more often than usual. Some experiments on hyperparameters were performed nonetheless, the details of which can be found in Appendix A. Quantized versions were used due to local hardware restrictions.

The model performed better with the thinking mode toggle on, and with the following hyperparameters: *sampling temperature* of *0.6*, *top-p* value of *0.95*, and a *top-k* value of *20*, as listed in its docs [9]. The model was also seen as being quite sensitive to changes in prompts, thus resulting in the adoption of multiple strategies to create the prompts as listed above. The 4-bit quantized version using BitsAndBytesConfig was the model that was used for reporting all these scores [10].

To support faster iteration, economical usage of energy and better examination of the thinking mode output, Ollama was used when crafting the prompts, with its application[2] as the backend and the accompanying Python library as the one handling prompts and responses. This was especially useful in arriving at the system and user prompts for the relation *awardWonBy*, since, for this relation, even with all the current measures in place, there exists the tendency to collapse into repetitive and incomprehensible output. The full system when run on a Quadro RTX 6000 GPU or similar takes around 2 hours to produce full results, with all the hyperparameters mentioned above in use.

4. Results

The outputs were evaluated on the basis of macro F1 scores for each relation, the details of which are in the tables below. Table 4 shows the results of the baseline system evaluated on test data, and Table 5 shows the results of this system on test data.

Table 4
Evaluation Results - Baseline - Test Set

Relation	macro-p	macro-r	macro-f1
awardWonBy	0.240	0.090	0.117
companyTradesAtStockExchange	0.185	0.591	0.167
countryLandBordersCountry	0.768	0.812	0.702
hasArea	0.240	0.240	0.240
hasCapacity	0.040	0.040	0.040
personHasCityOfDeath	0.080	0.650	0.080
All Relations	0.227	0.435	0.212

Table 5
Evaluation Results - System - Test Set

Relation	macro-p	macro-r	macro-f1
awardWonBy	0.191	**0.159**	**0.139**
companyTradesAtStockExchange	0.178	**0.604**	**0.170**
countryLandBordersCountry	0.737	**0.817**	0.690
hasArea	**0.260**	**0.260**	**0.260**
hasCapacity	**0.150**	**0.150**	**0.150**
personHasCityOfDeath	**0.090**	**0.660**	**0.090**
All Relations	**0.249**	**0.469**	**0.240**

[2] https://ollama.com/library/qwen3

5. Discussion

Examining the results, we can see that this system does beat the baseline, but barely so. Suffice to say, while these prompts work, there are definitely limits to what the LLM knows. Hence we can say that while the model may be sensitive to prompts themselves, there seems to be a limit to the world/relational knowledge it has, at least the way it is retrieved with these styles of prompts. This becomes especially clear in the case of the relation *countryLandBordersCountry*, where it becomes apparent that the baseline question prompt works better than the detailed prompt that this system proposes.

An area that the proposed system seems to be consistently better at is in retrieving the relevant elements, as can be seen from the recall scores, which are higher than the baseline in all cases. Nevertheless, in improving the baseline by only 2-3 percentage points, this work points to the fact that the baseline in itself is a strong one, while also leading to the conclusion that other, prompt-optimisation based methods need to be looked into to improve performance.

The following section details other experiments that were performed during the course of this study, but not kept in the final system due to various constraints on performance and other factors. They can serve as a starting point for further study on the same, if need be.

6. Experiments in multi-step prompting and quantized versions

Throughout the course of iterating over different versions of prompts for all relations, the outputs of the *awardWonBy* relation showed little to no improvement. This is similar to results from previous editions of the challenge where outputs for this relation performed significantly worse than others [5]. Moreover, for this system, for certain subjects in this relation, the outputs were repetitive and at times, incomprehensible and not following the specified format. Even after looking at the thinking mode output and creating prompts that explicitly prohibited these behaviours, they kept recurring.

Thus, in order to try and fix that, another idea that was explored was to use a second set of prompts after the first, instead of manual post-processing steps. The inspiration for this stems from previous works dealing with LLM-as-a-judge, dual prompting, and Prompting as Probing (ProP) methods that employ similar ideas in asking the model to check its output [6][11]. Since Qwen3-8B was the only model that could be used, the idea was to take the outputs from the system above, and feed it again to the same model under the premise that, given this subject-predicate-object triple furnished by an LLM, its duty was to correct the object if needs be and output it, all while following the required format.

Table 6
Evaluation Results - Twice Prompted - Validation Set

Relation	macro-p	macro-r	macro-f1
awardWonBy	0.044	0.004	0.008
companyTradesAtStockExchange	0.210	0.545	0.198
countryLandBordersCountry	0.616	0.878	0.614
hasArea	0.240	0.240	0.240
hasCapacity	0.040	0.040	0.040
personHasCityOfDeath	0.160	0.610	0.160
All Relations	0.224	0.425	0.221

The results of this approach on the validation dataset can be found in Table 6. A closer look reveals that while it does still beat the baseline, the performance of the relevant relation that we wanted, *awardWonBy*, remains subpar. Thus, this approach was not pursued further as it was not deemed efficient considering the time it takes to prompt the model twice.

Another path explored was to look at the 8-bit quantized version of Qwen3-8B instead of the 4-bit used above, to examine if there are pronounced performance gains. The results on the validation dataset can be found in Table 7. Again, this direction was not pursued further as well, as the results were found to be not that different from what the 4-bit quantized version produces. Hence, the decision was made to stick with the 4-bit quantized Qwen3-8B model for reasons of economy and speed. All quantized versions were deployed using the BitsAndBytes framework [10].

Table 7
Evaluation Results - 8-bit quantized Qwen3-8B - Validation Set

Relation	macro-p	macro-r	macro-f1
awardWonBy	0.170	0.058	0.073
companyTradesAtStockExchange	0.225	0.573	0.217
countryLandBordersCountry	0.636	0.916	0.645
hasArea	0.230	0.230	0.230
hasCapacity	0.080	0.080	0.080
personHasCityOfDeath	0.080	0.530	0.080
All Relations	0.223	0.427	0.220

Further experiments featuring various combinations of hyperparameters such as variations in temperature and the number of examples, as well as toggling between think and no-think modes, is available on Appendix A.

7. Future work

Apart from the above, one of the other directions this work can take includes benchmarking it and comparing it to the winning system from last year, which was a Llama3-8B-Instruct model prompted with specially designed prompts and with an entity disambiguation step at the end [5]. By adding entity disambiguation to the predicted objects and ensuring uniform prompt designs, both models can be compared side by side as open-weight 8B models, although for a more accurate comparison, it would be wise to take an instruction-tuned Qwen3 or a non instruction-tuned LLama3.

Another direction is to look at prompt-tuning methods for better results. Since they involve using learnable parameters in the input embeddings in order to design the best prompts automatically, better results than the baseline can be expected. A look at the other systems in this challenge will provide insight into the same. Finally, the effects of reintroducing entity disambiguation for the predicted objects, as in previous years, can also be studied to see how it improves the results.

8. Conclusion

This work explored how a 4-bit quantized Qwen3-8B model can be used for the knowledge base completion task using manually designed prompt templates alone. It showed how the baseline system is itself a strong one, and how certain multiple-object relations like *awardWonBy* remain challenging for LLMs to solve. Qwen3-8B as a model with the thinking mode toggle on remains highly dependent on its hyperparameters for the generation of accurate and relevant output, even for this task. And while the prompt templates furnished with this work can be used as a baseline, better results seem possible only if prompt-optimisation, disambiguation, and retrieval-augmented methods are studied and employed.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] J. Kalo, T. Nguyen, S. Razniewski, B. Zhang, Lm-kbc challenge @ iswc 2025, in: 4th Semantic Web Challenge on Language Models for Knowledge Base Construction Challenge, 2025. URL: https://lm-kbc.github.io/challenge2025/.

[2] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language models as knowledge bases?, 2019. URL: https://arxiv.org/abs/1909.01066. arXiv:1909.01066.

[3] A. Nayak, H. P. Timmapathini, LLM2KB: Constructing Knowledge Bases using instruction tuned context aware Large Language Models (2023). URL: https://arxiv.org/abs/2308.13207. doi:10.48550/ARXIV.2308.13207, publisher: arXiv Version Number: 1.

[4] Y. Hu, T.-P. Nguyen, S. Ghosh, S. Razniewski, Enabling LLM Knowledge Analysis via Extensive Materialization (2024). URL: https://arxiv.org/abs/2411.04920. doi:10.48550/ARXIV.2411.04920, publisher: arXiv Version Number: 4.

[5] D. M. R. Bara, Prompt engineering for tail prediction in domain-specific knowledge graph completion tasks (2024). URL: https://ceur-ws.org/Vol-3853/paper10.pdf.

[6] A. Das, N. Fathallah, N. Obretincheva, Navigating Nulls, Numbers and Numerous Entities: Robust Knowledge Base Construction from Large Language Models (2024). URL: https://ceur-ws.org/Vol-3853/paper12.pdf.

[7] HuggingFace Docs - PEFT, PEFT, 2025. URL: https://huggingface.co/docs/peft/main/en/index.

[8] HuggingFace Docs - PEFT, PEFT - Prompt-based methods, 2025. URL: https://huggingface.co/docs/peft/main/en/task_guides/prompt_based_methods.

[9] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, Z. Qiu, Qwen3 Technical Report, 2025. URL: http://arxiv.org/abs/2505.09388. doi:10.48550/arXiv.2505.09388, arXiv:2505.09388 [cs].

[10] T. Dettmers, L. Zettlemoyer, The case for 4-bit precision: k-bit inference scaling laws, in: International Conference on Machine Learning, PMLR, 2023, pp. 7750–7774.

[11] D. Alivanistos, S. B. Santamar'ia, M. Cochez, J.-C. Kalo, E. v. Krieken, T. Thanapalasingam, Prompting as Probing: Using Language Models for Knowledge Base Construction, ArXiv (2022). URL: https://www.semanticscholar.org/paper/ddc9aeac18638575bbb90ede4c6829ec15c2947e.

A. Experiments on hyperparameters

This section details the results of the various experiments that were performed, reported on the validation set. Apart from the hyperparameter that is being modified, these are the configuration settings that were followed all throughout in the following experiments, unless stated otherwise:

1. LLM - 4-bit quantized Qwen3-8B with thinking mode on
2. prompts / templates - as in the final system submitted
3. *max_new_tokens* - 4096
4. *temperature* - 0.6
5. *top_k* - 20
6. *top_p* - 0.95
7. *min_p* - 0
8. *few_shot* - 5

Among experiments that were not run due to local hardware restrictions are the system on the full, non-quantized model, and the system with the full, 32k context, the first due to memory constraints, and the latter primarily due to time constraints (at $\sim$6 hours, the model was still only 42% done).

When comparing the performance of the system in Table 8(A) with that of Table 10(B), we can see that the macro f1 scores are quite similar, with the latter edging out the former especially in cases of the relations like *awardWonBy* and *companyTradesAtStockExchange*, and in the case of overall recall. The margins are negligible though and the performance comparable, hence, the system with *max_new_tokens* - 4096 was kept as the other took twice longer to run per full set.

Similarly, *awardWonBy* and *companyTradesAtStockExchange* also seem to benefit from the settings in Table 8(B) and Table 9(A), i.e., with the thinking mode toggle off, or the *temperature* at 0. *awardWonBy* benefits from longer contexts as well, as seen in Table 10. This is quite surprising, as, at the same time, the same relation seems to benefit from both thinking mode, as well as a *temperature* at 0 setting. Looking further at the specific outputs may be able to provide insight into what is happening in this case.

Table 8
Evaluation Results - System, and System with Thinking Mode Off - Validation Set

	(A) System			(B) System with Thinking Mode Off		
Relation	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
awardWonBy	0.145	0.032	0.047	0.176	**0.046**	0.057
companyTradesAtStockExchange	0.215	0.565	0.211	0.220	**0.629**	0.237
countryLandBordersCountry	**0.648**	0.891	**0.638**	0.559	0.895	0.585
hasArea	0.230	0.230	0.230	0.140	0.140	0.140
hasCapacity	0.100	0.100	0.100	0.090	0.090	0.090
personHasCityOfDeath	**0.130**	**0.580**	**0.130**	0.100	0.550	0.100
All Relations	**0.236**	0.436	**0.232**	0.198	0.423	0.203

Table 9
Evaluation Results - Model Temperature at 0, and Zero Shot - Validation Set

	(A) Temperature 0			(B) Zero Shot		
Relation	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
awardWonBy	0.179	0.041	0.058	0.092	0.023	0.036
companyTradesAtStockExchange	0.223	0.578	0.215	0.168	0.518	0.165
countryLandBordersCountry	0.608	0.897	0.610	0.499	0.768	0.501
hasArea	0.220	0.220	0.220	**0.240**	**0.240**	**0.240**
hasCapacity	0.090	0.090	0.090	0.050	0.050	0.050
personHasCityOfDeath	0.050	0.500	0.050	0.080	0.530	0.080
All Relations	0.212	0.419	0.208	0.186	0.390	0.184

Table 10
Evaluation Results - *max_new_tokens*: 8192 and 16384 - Validation Set

	(A) *max_new_tokens*: 8192			(B) *max_new_tokens*: 16384		
Relation	macro-p	macro-r	macro-f1	macro-p	macro-r	macro-f1
awardWonBy	0.143	0.045	0.056	**0.183**	0.044	**0.060**
companyTradesAtStockExchange	0.192	0.533	0.181	**0.249**	0.618	**0.248**
countryLandBordersCountry	0.612	0.888	0.616	0.637	**0.902**	0.637
hasArea	0.210	0.210	0.210	0.200	0.200	0.200
hasCapacity	0.060	0.060	0.060	**0.110**	**0.110**	**0.110**
personHasCityOfDeath	0.090	0.540	0.090	0.100	0.550	0.100
All Relations	0.205	0.408	0.202	0.232	**0.438**	0.229

Overall, as visible from above, different settings seem to have different strengths, as visible from the fact that, despite its poor overall performance, the zero shot setting in Table 9(B) produces the best results for the relation *hasArea*. Thus, to choose a well-balanced system or not will depend on the use-case as well as the input data fed into the model. Here at least, the hyperparameters as provided in the model docs seem to provide the best, well-balanced output [9].

B. System prompts

One of the system prompts for the system proposed in this paper is as follows:

> You are an expert at quick and accurate knowledge base completion. Your expertise is in the following: Given a subject and a relation, you provide the correct object(s) to complete that triple, as a comma-separated list. The object may be a null object, or a single object, depending on the subject and the relation provided. If you know a part of the answer, you will provide the part you know, and ONLY if you do not know the answer do you type None.
>
> Here are the rules you must follow for this specific task: Penalties for: 1. Repeating the object more than once for a specific subject and relation - there is a heavy penalty for repetition! 2. Providing more information than the object(s) Hence 1 and 2 should be avoided at all costs. If your answers are too long, limit yourself to the first 200 objects per triple. If the user provides any hints or examples, you may take those into account to help you. You do not think whether the answer is long or short or unhelpful, you only focus on the correctness of the answer. You do not get caught up in the semantics of specific words, but rather focus on the subject and the relation. You are brusque and to the point, and you only provide the object, nothing else.
>
> Ways to approach this task: - Try to place the subject in context - be it geographically, or in terms of its specific field or activity, or, in the case of a person - their life and work. Once you have identified the subject, try and verbalise the relation, then think of what type of object(s) would fit that relation. You can then proceed to find the specific object(s) that would fit that relation. - Alternatively, after identifying the subject, you can construct your own knowledge base by listing everything you know about the subject. Then, considering the relation, you can find the object in the knowledge base you just constructed.
>
> The user will evaluate your answers based on the correctness of the object(s) you provide, so accuracy and precision is key, and remember, no repetitions. Finally, let me remind you of the most important rule: You will not provide any explanation, just the object(s) in a comma-separated list.

In the first paragraph, the expertise is declared, as well as the type of input and output expected. Also declared are possibilities of a *null* answer, so as to obtain output in the desired format in such a case. The second paragraph details the rules and general ideas that the model should follow while the third gives it ideas on how to approach the task. The fourth and final paragraph closes out by reiterating both the points that should be kept in focus, as well as the expected output format.

The user prompts below follow a similar idea in that, it details what is unnecessary and not to be included in the output, as well as detailing ways to think about the question at hand. For example, in the case of *hasArea*, the prompt states that units are not to be displayed, just the numeric value. And in the case of *companyTradesAtStockExchange*, the prompt asks to keep track of the stock ticker for a company per stock exchange so as to not lose the information while on thinking mode. Further details can be found in the prompts below.

C. User prompts

The following are some of the user prompts in use for this system:

hasArea - State the geographical area of {subject} in square kilometres, without printing the unit. Locate it geographically, identify the kind of geographical feature or entity it is, and then find its area. Your sources could be wikipedia, encyclopedias, physical maps, guidebooks and so on. If you are unsure, (1) think about similar subjects and their area, or (2) features nearby {subject} to help you find the area which is the object. Three things to note: 1. The area is in square kilometres, so if the area you find is in a different unit, convert it to square kilometres before giving it to me, but do not print the unit. 2. If for some reason you have conflicting information, say for example, two or more sources that give you different areas, then use the most reliable source you have. 3. If you have multiple subjects with the same name, then provide the area for the most well-known subject with that name. Here you go! Subject: {subject}, Relation: {relation}, Object: ?

hasCapacity - Give me the capacity of {subject} in number of people, but do not print the unit. If you are not sure about the exact capacity, provide the best estimate you can by thinking of similar subjects across the world, or other subjects in the same geography. Consider the {subject}, its geographical location and its capacity. If you can find sources like wikipedia or news articles that list the capacity of {subject}, those would be your best bet. You could also identify the purpose that {subject} is used for, and find its capacity by taking into account similar structures and the geography that {subject} is located in. Here you go! Subject: {subject}, Relation: {relation}, Object: ?

awardWonBy - Here, you should list the names of the objects(awardees/winners) who have won the {subject} award/honor. The awardees of {subject} could be persons, specific entities (like album names or band names in the case of music, or entire teams from countries in the case of team sports for example) or organisations. Consider the {subject}, the field in which it is given, and its periodicity (for instance, awarded early or not). List as many awardees as you know, or try to provide a list of famous recipients at least. Do not repeat the objects/awardees though. {subject} is an award/honor that exists, so try and match the award exactly – not to something similar, but that exact award. Try to find sources like news or reports that talk about these awards and awardees. List years and awardees as an intermediate step so as to keep track, and them combine it all together. And remember, even if a person received it multiple times, I want them listed only once. Having no repetitions is key. Here you go! Subject: {subject}, Relation: {relation}, Object: ?

companyTradesAtStockExchange - Here, list the stock exchanges where {subject} trades at present. Remember, objects are stock exchanges and stock exchanges alone. Consider the company, its headquarters and locations, the indices it is part of and the stock exchanges it is listed at. If you can find, and are sure that {subject} trades at specific stock exchanges, list those stock exchanges. It could be that {subject} trades at stock exchanges in geographies other than where it is headquartered, so do not limit yourself to just one geography. And in the same country too, there could be multiple stock exchanges where the {subject} is listed at. Remember, a company may trade at multiple stock exchanges across geographies, so apart from annual company reports and the like, you can use financial news and reports to find your answers. No guessing - I want you to try and verify your answers by finding the relevant stock ticker/code at the related stock exchange as well (because one ticker/code per exchange), but do not list the tickers/codes, only the name of the stock exchanges. If you are not sure about the stock exchange, or if you are sure the company is not listed at all, return None. Here you go! Subject: {subject}, Relation: {relation}, Object: ?