GECKO: A Question Answering System for Official Statistics

Lucas Lageweg[1,2], Jonas Kouwenhoven and Benno Kruit[3]

[1]Statistics Netherlands, Henri Faasdreef 312, Den Haag, Netherlands
[2]University of Amsterdam, Science Park 900, 1098 XH Amsterdam, Netherlands
[3]Vrije Universiteit Amsterdam, De Boelelaan 1105, Amsterdam, Netherlands

Abstract

This paper presents GECKO, a knowledge graph-based statistical question answering system currently in beta deployment. GECKO aims to facilitate the retrieval of single statistical values from an extensive database containing over a billion values across more than 4,000 tables. The system integrates a comprehensive framework including data augmentation, entity retrieval, and large language model (LLM)-based query generation. A key feature of the beta deployment is the collection of user feedback, which is critical for improving system performance and accuracy. This feedback mechanism allows users to report issues directly, ensuring continuous improvement based on real-world use.

1. Introduction

Statistics Netherlands (Centraal Bureau voor de Statistiek; CBS) is an independent administrative body of the Dutch government tasked with the creation of statistics over a broad spectrum of social topics and the responsibility to make them accessible to the general public. However, in-house studies have shown that users struggle to find the correct tables for their needs in the vast amount of data available. This research aims to develop a Question Answering (QA) system to provide specific statistical observations from this data as responses to natural-language user questions.

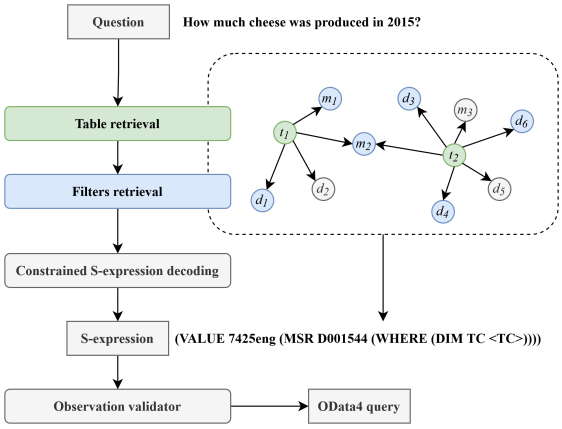
QA systems can take several forms, with most recently free-form generative Large Language Models (LLMs) like ChatGPT and GPT4 [1] getting much attention. Due to the nature of these models, they are able to generalize very well on a large range of topics, but have shown to be prone to 'hallucinations', where plausible but incorrect or even nonsensical answers are generated [2]. Especially for official data like governmental statistics, this is highly undesirable behavior.

Knowledge Graph Question Answering (KGQA) is a field where knowledge graphs (KGs) containing real-world facts and relations in structured form are used as a basis for QA systems. Answers of such systems should always adhere to the KG. Therefore, assuming it contains correct information, answering by returning parts of the KG, or reasoning over it, cannot lead to nonsensical answers. In this paper, we introduce an end-to-end pipeline for a generation-based KGQA system of CBS data.

SEMANTICS 2024, Demo Track
l.lageweg@cbs.nl (L. Lageweg); jonaskouwenhoven@live.nl (J. Kouwenhoven); b.b.kruit@vu.nl (B. Kruit)

Our approach introduces a data augmentation process for enhancing model training, explores various encoder architectures for entity retrieval, and proposes a new query generator mechanism enhanced by Low Rank Adaptation (LoRA) [3]. Additionally, we propose a new prompting technique that utilizes dynamic prompts, constructing specific prompts based on the generation phase. These improvements help the process of generating symbolic expressions for querying a KG, thereby enhancing the overall performance of the QA system.

This paper details the beta deployment of GECKO and its feedback collection mechanism, emphasizing the role of user input in refining the system. The beta phase is critical for identifying and addressing potential issues, ultimately enhancing the system's robustness and reliability.



(a) Overview

Periods	Dairy production	
	Butter	Cheese
	1 000 kg	
1995	132,300	682,900
2000	126,200	683,600
2005	118,800	672,200
2010	133,419	752,638
2015	147,577	844,974

(b) Example Table Fragment

Figure 1: (a) Overview of our pipeline from query to answer. Candidate table nodes (green) are retrieved for the query, after which the measure and dimension filter candidates are retrieved (blue), resulting in a complete subgraph for the table candidates. The subgraph is used as input for the constrained S-expression decoding by either the baseline method or trained model. (b) Example CBS table fragment (from `7425eng`), showing one dimension (time periods) and two measures.

2. Related Work

Query generation systems, particularly those involving text-to-SQL and KGQA, have made significant strides [4]. Recent work [5, 6] focus on grounding queries in knowledge graphs to avoid hallucinations. Recent advancements [7, 8] highlight the use of LLMs in generating logical forms for querying databases.

Data augmentation techniques [9, 10], are essential for creating diverse and realistic training datasets. Entity retrieval methods including sparse and dense retrieval approaches [11, 12, 13], play a crucial role in identifying relevant data within vast datasets.

Compared to existing KBQA or text-to-SQL systems, we provide a hybrid solution where statistical tabular data can be represented as knowledge graphs, to which the techniques for symbolic expression generation instead of more complex query language generation (SQL or SPARQL) can be applied. With this approach, we propose a novel system that can help find

relevant information in official statistics and similar systems, which is vital for governmental decision making and all fields of research utilising and relying on these statistics.

3. System Design

In processing a question, GECKO performs four core steps: entity retrieval, filters retrieval, constrained S-expression decoding (i.e. symbolic expression generation) and observation validation. We restrict the querying space by performing entity retrieval based on the input question to determine the closest KB nodes. This is done through either sparse retrieval using BM25+ [14] as a baseline method, using a trained dual encoder [12] or a finetuned ColBERT model [13].

After obtaining the closest matching entities based on the query, we retrieve all possible filters for tables by exploding a subgraph using the entities found. The result of the subgraph exploding is a graph containing all table nodes and their related measures and dimensions having nodes intersecting with the retrieved entities from the previous step. The subgraph contains all relevant nodes to the query, connected to one or more tables.

The query and subgraph are used as input for the constrained S-expression decoding. The S-expressions are generated token-by-token such that, given the subgraph, admissible tokens are returned at every step. A rule-based baseline was created using the entity retrieval scores to greedily determine what token from the admissible tokens to select. The second method uses a transformer-based decoder-only seq2seq model and dynamic prompting.

When generating a token at a given timestep, the model evaluates the sequences in the list of admissible/constrained tokens and selects the sequence with the highest assigned score. For example, when `7425eng` is given to the decoder as one of the admissible next tokens, but only a decomposition of sub-tokens can be embedded by the model (e.g. `7425` followed by `##eng`), the summed log probability for these subtokens will determine the total probability of selecting this identifier for generation.

The novelty in this constraining method is the introduction of *dynamic prompting*, which, instead of calculating the likelihood of a token sequence based on a static prompt (i.e. text input for the model), adjusts the prompts according to the generation phase. For example, when generating a table ID, the prompt is altered to only include the most relevant table IDs and their descriptions. Similarly, when measures are generated in the next phase, it retrieves the measures related to the previously generated table ID, and using those to construct a new prompt. This method applies to the different dimension groups as well. Figure 2 contains a schematic overview of the dynamic prompting technique.

4. Model training & beta deployment

For creating training data, we developed a method for manual data annotation. This method involves annotators writing queries that can be answered by a specific table cell. Annotators were instructed to write their questions both as full sentences and in a more casual style, aiming to simulate the formulation of questions posed by users in a search engine. The data obtained from this manual annotation process contains queries and their corresponding S-expression, resulting in 2300 annotated pairs.

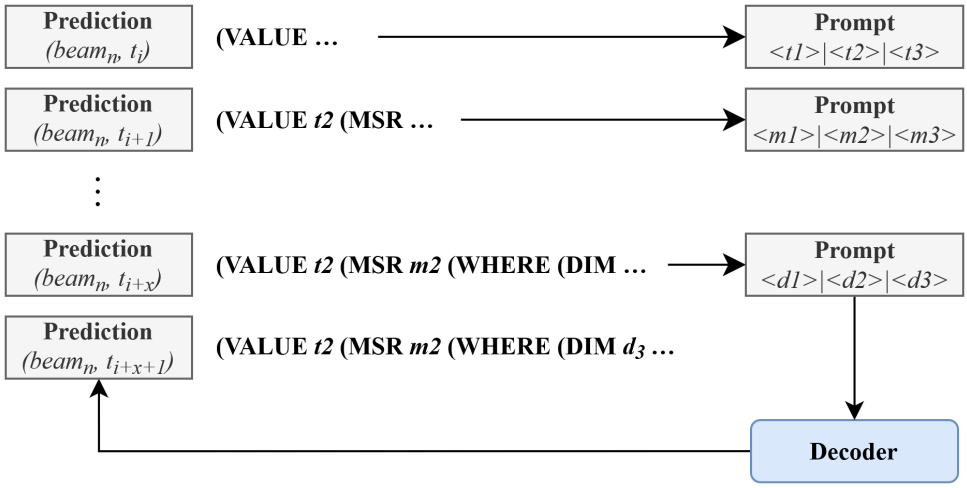


Figure 2: Schematic overview of dynamic promting for a decoder-only model architecture. If there are multiple tokens that can be generated, a custom prompt is given to the decoder with labels for all the possible options.

The annotated queries were distributed over random tables from the CBS datapool, and contained a strong class imbalance towards tables that were more easily annotated. This class imbalance and random distribution motivates extending this study with data augmentation. In this extension, annotated S-expressions and their associated queries are used to fine-tune a GPT-3.5 model through the OpenAI fine-tuning services. The query-expression pairs were transformed into prompts using the descriptions of the IDs for various measures, dimensions, and table IDs. Training such a model reduces the need for additional manual annotation, while also significantly increasing the amount of annotated data.

The initial GECKO model (v_1) and model containing the improvements discussed here (v_2) were evaluated using a selected sample of this dataset. This was done by evaluation exact table matches of generated S-expressions (v_1 0.35; v_2 **0.63**), F1-scores for selected dimensions in said S-expressions (v_1 0.62; v_2 **0.71**) and by manually annotation answer relevancy, as an answer can be a non-exact match but still be relevant to the question (v_1 0.38; v_2 **0.71**).

5. Conclusion

The beta deployment of GECKO[1], a generation-based KGQA system for CBS data, marks a significant milestone in improving user interaction with governmental statistics. This phase includes mechanisms for feedback collection, which will play a crucial role in refining and enhancing the system based on user input. The feedback gathered during the beta deployment will help identify and address potential issues, ensuring the system's robustness and reliability. This process is essential for developing a reliable QA system capable of providing accurate and relevant statistical observations in response to natural-language user questions.

[1]https://gecko.cbs.nl

References

[1] OpenAI, GPT-4 Technical Report, 2023. arXiv:2303.08774.

[2] M. Zhang, O. Press, W. Merrill, A. Liu, N. A. Smith, How Language Model Hallucinations Can Snowball, 2023. arXiv:2305.13534.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations, 2022.

[4] B. Qin, B. Hui, L. Wang, M. Yang, J. Li, B. Li, R. Geng, R. Cao, J. Sun, L. Si, F. Huang, Y. Li, A survey on text-to-sql parsing: Concepts, methods, and future directions, 2022. arXiv:2208.13629.

[5] Y. Gu, S. Kase, M. Vanni, B. Sadler, P. Liang, X. Yan, Y. Su, Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases, in: Proceedings of the Web Conference 2021, ACM, 2021. doi:10.1145/3442381.3449992.

[6] D. Yu, S. Zhang, P. Ng, H. Zhu, A. H. Li, J. Wang, Y. Hu, W. Wang, Z. Wang, B. Xiang, Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases, 2023. arXiv:2210.00063.

[7] P. Schneider, M. Klettner, K. Jokinen, E. Simperl, F. Matthes, Evaluating large language models in semantic parsing for conversational question answering over knowledge graphs, arXiv preprint arXiv:2401.01711 (2024).

[8] L. Nan, Y. Zhao, W. Zou, N. Ri, J. Tae, E. Zhang, A. Cohan, D. Radev, Enhancing text-to-SQL capabilities of large language models: A study on prompt design strategies, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.

[9] L. Bonifacio, H. Abonizio, M. Fadaee, R. Nogueira, InPars: Unsupervised dataset generation for information retrieval, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 2387–2392. doi:10.1145/3477495.3531863.

[10] V. Jeronymo, L. Bonifacio, H. Abonizio, M. Fadaee, R. Lotufo, J. Zavrel, R. Nogueira, Inpars-v2: Large language models as efficient dataset generators for information retrieval, arXiv preprint arXiv:2301.01820 (2023).

[11] Z. Huang, S. Xu, M. Hu, X. Wang, J. Qiu, Y. Fu, Y. Zhao, Y. Peng, C. Wang, Recent trends in deep learning based open-domain textual question answering systems, IEEE Access 8 (2020) 94341–94356.

[12] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.

[13] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, in: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 2020, pp. 39–48.

[14] Y. Lv, C. Zhai, Lower-bounding term frequency normalization, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11, Association for Computing Machinery, New York, NY, USA, 2011, p. 7–16. doi:10.1145/2063576.2063584.