Entity Examples for Explainable Query Target Type Identification with LLMs [⋆]

Darío Garigliotti$^{1[0000-0002-0331-000X]}$

University of Bergen, Norway - dario.garigliotti@uib.no

Abstract. When answering a user query with relevant entities from a knowledge base (KB), utilizing their semantic class or type information typically structured in the KB is known to improve the retrieval performance for these entities. Accordingly, it is important to identify the target types of entities expected by a query. This work addresses the task of Target Type Identification (TTI) by replacing the established supervisedly learnt ranking approach with a generative approach powered by Large Language Models (LLMs). Beyond assessing the ability of LLMs at predicting query target types, we study aspects of the strategy to elicit generation, in particular, the role of example relevant entities in supporting the explanation of mechanisms behind the LLM predictions.

Keywords: Target Type Identification · Entity Retrieval · LLMs.

1 Introduction

When searching on the Web, users often issue queries that are better answered with entities, such as people, locations, organizations and events. For example, the query *'nobel prize winners physics'* aim for answers including entities such as Marie Curie and Albert Einstein. This entity-centric paradigm has consolidated in major commercial search engines, since results are no longer just "blue links" pointing to documents with relevant information but instead direct results via widgets and cards with entities as first-class citizens (Balog, 2018). Millions of entities within multiple domains of knowledge, are typically uniquely identified in a knowledge base (KB) of reference to the search engine that stores structured information about them. The mentioned answer entity Albert Einstein, in a typical KB (with, for simplicity, unique identifier `Einstein`), is possibly related with other entities (e.g. ⟨`Einstein`, *born_ in*, `Ulm`⟩) and with a distinguished unit of information, its semantic class or type (e.g. ⟨ `Einstein`, *is_ a*, `Physicist`⟩). In the problem of entity retrieval (ER) (this is, returning a list of entities from a KB for an input query such that they are ordered by relevance) (Balog et al., 2011, 2012), it is known that incorporating the *target types* of the query (i.e. the types of its relevant entities) can improve the performance of ER methods (Garigliotti et al., 2019). Hence, in this paper we address the problem of query *target type identification* (TTI) (this is, returning a ranked list of target types for an input

[⋆] Part of NRF project 329745 Machine Teaching For XAI.

query) (Garigliotti et al., 2017), which is fundamental to obtain the type-based information to be incorporated during ER.

Established TTI approaches rely on a learned ranking method that aggregates complementary supervised learning features, including query attributes, type attributes, and features capturing the relevance of a type as a target for a query based on underlying suboptimal ranking methods (Garigliotti et al., 2017). Representation -or deep- learning (DL), instead, allows for abstracting the feature design altogether and learn the feature set itself. With the notable DL-driven development in Language Models, especially the recent state-of-the-art performance by Large Language Models (LLM) in a variety of tasks (Radford et al., 2019; Si et al., 2023), we are interested in assessing the ability of LLMs for addressing TTI, while also making the predictions that it generates explainable. This aligns with the increasing demand in the research community for developing trustworthy technologies for an also increasing number of human-centered LLM applications (Liu et al., 2023). Specifically, we aim to exploit the relevant entities of a query, as natural bridges with its target types, by including them in the prompt when asking an LLM to generate these types. We expect these entities to serve as witnesses to explain the rationale behind the particular target type outputs generated by the LLM. Our approach brings together these different information units (query, entities, types) within a series of methods under Retrieval-Augmented Generation (RAG), a framework that extends the implicit information stored in the model parameters with explicit knowledge incorporated while prompting an LLM (Garigliotti et al., 2024; Garigliotti, 2024). This paper presents initial developments on assessing the impact of these entities in serving as witnesses to explain the rationale behind the particular target type outputs generated by the LLM.

2 Methodology

2.1 Problem

Given an entity-oriented query q —i.e. a query whose expected answers are all entities— and a type hierarchy T of reference —typically available in a KB—, the task of *Hierarchical Target Type Identification (TTI)* (Balog and Neumayer, 2012; Garigliotti et al., 2017) consists in returning a ranked list $R(q) \in T$ such that $R(q)$ has all the main target types of the query, this is, such that (i) they are the most specific category of entities that are relevant to the query, and (ii) they are not on the same path from the root in the tree induced by T. In particular, (ii) means that if $t \in R(q)$ and t is a descendant of t' in T, then $t' \notin R(q)$.

2.2 Approach

We approach LLM-based TTI via a series of methods all within the common framework of Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Gao et al., 2023). Specifically, in our instantiation of the RAG pipeline, the first stage, retrieval, obtains relevant target types for the input query in a first, solid pass. The second stage, augmentation, incorporates these retrieved items in a well-engineered prompt that requests to identify the target types among the candidates, according to the TTI problem specification also provided in the prompt. In the final stage, generation, an LLM is prompted to answer the question requiring to solve the TTI task. Starting from a basic prompt that contains the query and type candidates from which the LLM must identify the correct ones, we experiment with alternative configurations where increasing sets of relevant entities are also provided. We are interested in how these entity sets can be understood to explain an LLM generation prompted with relevant entity examples. Our experiments are designed and conducted to answer these research questions:

- **RQ1**: How does RAG perform when assuming (near-)optimal retrieval stage?
- **RQ2**: What is the impact of the order of types, and of few-shot illustrations?
- **RQ3**: How does the number of entity examples explain the generated types?

2.3 Experimental Setup

Datasets. DBpedia-Entity (Balog and Neumayer, 2013) is a test collection for entity retrieval comprising 485 queries compiled from different benchmarking campaigns, with their respective relevant entities from DBpedia 2015-10 KB judged by human annotators. The TTI test collection (Garigliotti et al., 2017) extends DBpedia-Entity with judgments on types from the DBpedia 2015-10 ontology for 479 queries (the missing 6 fail to have any meaningful type).

RAG configurations. In the **retrieval** stage, for simplicity, we assume that a perfect target type retriever is in place. This allows us to focus on aspects operationalized in the augmentation phase, especially the provision of entity examples, and on assessing how well an LLM identifies among these types during generation. We refer to the TTI test collection —itself an optimal retriever— as the *oracle*. We also consider a *pseudo-oracle*, this is, a near-optimal TTI method, which extends the oracle for each query with the union of the sets of types for all its entities retrieved with BM25 (Robertson, 1977), a solid lexical retrieval method. Then, during **augmentation**, we build a prompt that presents the query and retrieved type candidates, and asks to identify the correct target types as defined in Section 2.1. Alternative experiments also provide examples of entities relevant to the query. Orthogonally, we experiment with few-shot prompting with illustrations of the expected output answer for an input instance made of query, type candidates and possibly entity examples. Finally, at **generation** stage, we input the prompt into the GPT-3.5 (gpt-3.5-turbo-0125) LLM (Radford et al., 2019). This is a summary of our parameter configurations:

- Retrieval: **method** —oracle or pseudo-oracle—.
- Augmentation: **order** of the passages in the prompt —as retrieved in ranking, or random—; **number** of few-shot illustrations —0 or 1—; and **size** of the entity example set —0, 1, 2, 3, 4, or 5—.

Each possible combination of values set for all the experimental parameters determines a (RAG-based) TTI method in our study.

Evaluation metrics. We report the average performance across all the queries in TTI collection, measured in terms of precision, recall, and F-score.

3 Experimental Results

Answering RQ1: In all settings and for all metrics, the best performing methods use, as expected, optimal type candidate retrieval. The pseudo-oracle cases are more realistic scenarios where retrieval stage is imperfect, and the LLM is confused by the additional near-optimal type candidates.

Answering RQ2: In general, differences in performance by the order of provided types are very small when assuming oracle retriever. With pseudo-oracle, correct types appear (i) ranked from ground truth and (ii) before the additional candidates, so differences are larger and always favour the order by ranking. Results about the impact of zero- versus one-shot prediction are mixed.

Answering RQ3: The example set of size 1 provides only the highest-ranked entity, and increasingly larger sizes correspond to deeper entity retrieval cut-offs. We observe a very slight improvement in precision when adding more entities, and a substantial drop in recall when adding the first entity in zero-shot mode.

4 Conclusion and Future Work

In this work, we have studied the usage of LLM-powered Retrieval-Augmented Generation methods for query target type identification. This preliminary development sheds light on the impact of relevant entities as support when approaching explainability of the generation of predicted target types.

Ideally, the example entity set would be as small as possible while supporting optimal TTI performance. In future research, we aim to frame our approach within machine teaching (MT) (Zhu et al., 2018; Telle et al., 2019) —addressed only adjacently in our work—, whose formalism centers around such a minimization of the witness set used to train a machine learner for identifying concepts. A possible direction is to formalize it in terms of the target types as the concepts to be predicted by a learned model. Another space of research could explore, instead, the conceptualization by the underlying task of entity retrieval for a query. A third line of work would study further aspects of the TTI problem, such as (i) its hierarchical nature, (ii) the dedicated query grouping criteria in the test collection, as well as (iii) regarding the order between query-entity and entity-type bridging stages; dimensions that remain out of the scope of this preliminary work due to space limitations. Another missing side of the results is a qualitative analysis of which actual instances of queries and/or types are involved in every observed method.

Acknowledgments. This work was funded by the Norwegian Research Council grant 329745 Machine Teaching for Explainable AI.

Table 1: Experimental results for the studied methods over all the queries in the TTI test collection. In all these experiments, the generator LLM is GPT-3.5. In each block of this table, the best performance on a metric is shown in **bold**.

Retrieval method	**Types order**	**Zero-shot**			**One-shot**		
		Prec.	**Rec.**	**F-Sco.**	**Prec.**	**Rec.**	**F-Sco.**
Size of entity example set: 0							
Oracle	By ranking	**0.9687**	0.8341	0.8727	**0.9729**	**0.7119**	**0.7903**
	Random	0.9676	**0.8483**	**0.8831**	0.9541	0.6906	0.7692
Pseudo-O.	By ranking	0.8321	0.6971	0.7108	0.9207	0.6812	0.7521
	Random	0.8139	0.6785	0.6922	0.8768	0.6558	0.7209
Size of entity example set: 1							
Oracle	By ranking	**0.9812**	**0.7655**	**0.8305**	0.9791	0.7173	0.7955
	Random	**0.9812**	0.7587	0.8247	**0.9833**	**0.7223**	**0.8003**
Pseudo-O.	By ranking	0.8687	0.6814	0.7246	0.9217	0.6807	0.7523
	Random	0.8507	0.6623	0.7039	0.8768	0.6524	0.7191
Size of entity example set: 2							
Oracle	By ranking	**0.9833**	0.7566	0.8237	**0.9875**	**0.726**	**0.8043**
	Random	**0.9833**	**0.7679**	**0.8322**	0.9791	0.7166	0.7953
Pseudo-O.	By ranking	0.8723	0.6684	0.7209	0.9092	0.6724	0.7419
	Random	0.8441	0.6412	0.6942	0.8747	0.6494	0.7157
Size of entity example set: 3							
Oracle	By ranking	**0.9916**	0.7505	0.8219	0.9833	0.7196	0.7983
	Random	**0.9916**	**0.7536**	**0.8245**	**0.9854**	**0.7206**	**0.7997**
Pseudo-O.	By ranking	0.8673	0.6558	0.7129	0.9071	0.6694	0.7396
	Random	0.8403	0.6412	0.693	0.8601	0.6369	0.7031
Size of entity example set: 4							
Oracle	By ranking	0.9875	**0.7609**	0.828	**0.9833**	**0.7181**	**0.7971**
	Random	**0.9916**	0.7602	**0.8287**	0.9854	0.719	0.7984
Pseudo-O.	By ranking	0.876	0.6605	0.7203	0.904	0.6651	0.736
	Random	0.838	0.6407	0.6936	0.8674	0.6501	0.7143
Size of entity example set: 5							
Oracle	By ranking	**0.9875**	0.7574	**0.8254**	0.977	0.7133	0.7916
	Random	0.9861	**0.7581**	0.8242	**0.9854**	**0.7213**	**0.8003**
Pseudo-O.	By ranking	0.8753	0.6637	0.7209	0.9071	0.6692	0.7404
	Random	0.8252	0.627	0.6782	0.8685	0.6489	0.7141

References

K. Balog. *Entity-Oriented Search*, volume 39 of *The Information Retrieval Series*. Springer, 2018. ISBN 978-3-319-93933-9.

K. Balog and R. Neumayer. Hierarchical target type identification for entity-oriented queries. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2391–2394, 2012.

K. Balog and R. Neumayer. A test collection for entity search in DBpedia. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '13, pages 737–740, 2013.

K. Balog, M. Bron, and M. de Rijke. Query modeling for entity search based on terms, categories, and examples. *ACM Trans. Inf. Syst.*, 29(4):1–31, 2011.

K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the TREC 2011 Entity track. In *Proceedings of The 20th Text REtrieval Conference*, TREC '11, 2012.

T. Gao, H. Yen, J. Yu, and D. Chen. Enabling large language models to generate text with citations. In H. Bouamor and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore, Dec. 2023. Association for Computational Linguistics.

D. Garigliotti. SDG target detection in environmental reports using retrieval-augmented generation with LLMs. In *Proceedings of ClimateNLP, colocated with ACL*, pages 241–250. Association for Computational Linguistics, 2024.

D. Garigliotti, F. Hasibi, and K. Balog. Target type identification for entity-bearing queries. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 845–848, 2017.

D. Garigliotti, F. Hasibi, and K. Balog. Identifying and exploiting target entity type information for ad hoc entity retrieval. *Information Retrieval Journal*, 22(3):285–323, 2019.

D. Garigliotti, B. Johansen, J. V. Kallestad, S.-E. Cho, and C. Ferri. EquinorQA: Large Language Models for Question Answering over proprietary data. In *Proceedings of PAIS 2024*. IOS Press, 2024.

P. Lewis, E. Perez, and et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in NeurIPS*, volume 33, pages 9459–9474. Curran Associates, Inc., 2020.

N. Liu, T. Zhang, and P. Liang. Evaluating verifiability in generative search engines. In *Findings of EMNLP 2023*, pages 7001–7025. ACL, 2023.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019.

S. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

Q. Si, T. Wang, Z. Lin, X. Zhang, Y. Cao, and W. Wang. An empirical study of instruction-tuning large language models in Chinese. In *Findings of EMNLP 2023*, pages 4086–4107, Singapore, 2023. ACL.

J. A. Telle, J. Hernández-Orallo, and C. Ferri. The teaching size: computable teachers and learners for universal languages. *Mach. Learn.*, 108(8–9): 1653–1675, sep 2019. ISSN 0885-6125.

X. Zhu, A. Singla, and et al. An overview of machine teaching, 2018.