Towards Generative Semantic Table Interpretation

Viet-Phi Huynh[1], Yoan Chabot[1] and Raphaël Troncy[2]

[1]Orange, France
[2]EURECOM, France

Abstract
Semantic Table Interpretation (STI), or Semantic Table Annotation, is the process of understanding the semantics of tabular data with reference information identified in knowledge graphs (KG). In this paper, we first present insights gained from the design and implementation of DAGOBAH SL, a top performing STI system in state-of-the-art benchmarks, and we discuss the unsolved challenges that need to be addressed to make STI more effective in practice. Pre-trained generative Large Language Models (LLMs) have demonstrated their powerful versatility in tackling a broad spectrum of natural language understanding tasks. We envision their potential for improving STI systems. We describe several appealing research ideas that could lay the foundation for future development of Generative Semantic Table Interpretation.

Keywords
Semantic Table Interpretation, DAGOBAH, Knowledge Graph, Large Language Model, Generative Information Extraction

1. Introduction

Tabular data, as found in spreadsheets, is one of the most prevalent data structures used to store and present information on the Web and in industry [1, 2]. However, despite its compactness and human-readability, the lack of explicit semantics and the need for interpretation hinder its applications in dataset indexing, search and recommendation. To address this issue, Semantic Table Interpretation (STI) has emerged as an attractive research topic in recent years: this process aims at understanding the semantics of tables according to an ontology or a knowledge graph (KG), and at generating machine interpretable annotations. STI can typically refer to up to five tasks: i) Cell-Entity Annotation (CEA) disambiguates a cell mention by linking it to an entity in a KG such as Wikidata. For example, in Figure 1, the mention Arsenal is associated with the entity Q9617-Arsenal FC; ii) Column-Type Annotation (CTA) predicts semantic types for a column (e.g. concept Q476028-associated football club is assigned to column Col 0); iii) Columns-Property Annotation (CPA) represents the relationship between two columns using an existing property from a KG (e.g. Column Col 0 relates to column Col 2 via the relationship P15-home venue); iv) Topic Detection seeks to identify the subject conveyed by the table (e.g. in Figure 1, the table provides information on various football clubs-Q847017); v) Row-to-Instance reads an entire row and annotate it with an entity (e.g. the third row in Figure 1 represents the categorical entity Q8272924-Category: Aston Villa FC). This task is particularly useful in the case where a table has no clear subject column (see Figure 9 in [1]).

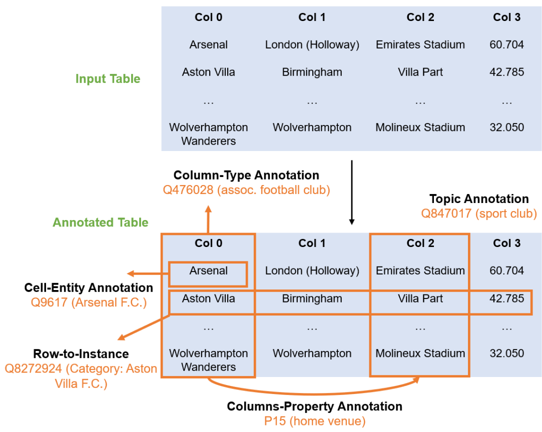

Figure 1: Illustration of five Semantic Table Interpretation tasks. The table is taken from the ToughTables [3] dataset.

The Semantic Web Challenge series on Tabular Data to Knowledge Graph Matching (SemTab [4, 5, 6, 7]) has fostered an increasing number of research works on STI topic. While various models have achieved excellent accuracy on most benchmark datasets, their applicability in practice is still under investigation, due to multiple challenges that remain, including the gap between the table data used in the benchmark and real world

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) — TaDA'23: Tabular Data Analysis Workshop, August 28 - September 1, 2023, Vancouver, Canada
*Corresponding author.
vietphi.huynh@orange.com (V. Huynh); yoan.chabot@orange.com (Y. Chabot); raphael.troncy@eurecom.fr (R. Troncy)
https://huynhvp.github.io/ (V. Huynh); https://yoanchabot.github.io/ (Y. Chabot); https://www.eurecom.fr/~troncy/ (R. Troncy)
0000-0001-7348-9650 (V. Huynh); 0000-0003-0457-1436 (Y. Chabot); 0000-0001-5639-1504 (R. Troncy)

scenarios, or the choice of algorithmic backbone [8, 9]. DAGOBAH SL [9, 10], the winning system of SemTab over the last two years, has shown to perform efficiently on *well-formed* relational tables (like the one in Figure 1) thanks to incorporating a rich set of match-based heuristics to evaluate the relevance between table context and entity graph. However, it still struggles with enterprise tables, heterogeneous tables that can be found in the wild Web or tables that have a low encyclopedic coverage (e.g. GitTables [11]).

This paper has two objectives. First, we present the insights gained when designing and implementing DAGOBAH SL, and we discuss the remaining scientific barriers that need to be tackled to obtain a reliable and generic annotation system. Second, in light of advancements in pre-trained generative Large Language Models (LLMs) [12] with emergent abilities and flexibility in solving a wide range of natural language understanding tasks, we envision several future steps we plan to experiment in leveraging LLM to fuel our table annotation system. Specifically, we rely on fine-tuning or few-shot learning techniques to adapt the model to table structure, inject and update knowledge within LLMs and use knowledge to generate auto-regressively the annotation in textual form.

2. DAGOBAH SL

DAGOBAH SL [9, 10] (SL standing for Semantic Lookup) is a framework for interpreting relational tables automatically via a two-stage pipeline: i) *preprocessing* is performed to clean the table and extract metadata, such as orientation, header, key column, column primitive typing (e.g. units, URL, email, etc). Importantly, this step can automatically detect {cells, columns, column pairs} targets that require annotation; ii) *annotation* follows a retrieve-then-rerank strategy where the retrieve phase searches for relevant KG entity candidates for a target cell mention via a keyword-based entity lookup, and then the rerank phase sorts out the most relevant entity for the cell (CEA task). The cell annotations are subsequently leveraged to predict the column types (CTA) and column-pair relations (CPA). The strength of DAGOBAH SL lies in two points:

A powerful keyword-based entity lookup service, built on Elasticsearch, indexing every label/alias of every entity in the alias table into an inverted index. It is capable of covering diverse surface forms of cell mentions such as acronyms, synonyms or typos through alias table enrichment, which results in a high recall within a few retrieved candidates. It incorporates three ranking factors: two similarity scores between mention and entity label/aliases calculated using edit distance and BM25, and a page-rank score quantifying the popularity of an entity. DAGOBAH SL currently supports various snapshots of DBpedia and Wikidata knowledge graphs.

An iterative cell entity/column type/column pairs relation disambiguation that leverages mutual interaction between table elements to optimize the re-ranking of candidate entities/column type/relations. For example, the types of a column can guide the ranking of entity candidates in cells associated with that column, and vice versa. The compatibility between table context and entity graph is evaluated thoroughly using a comprehensive set of matching rules including: i) semantic context plays more important role than literal context, ii) in the table, a neighboring column that is highly connected to target column should have higher contextual weight, iii) entity representation is expressed by a multi-hop graph centered around the entity, allowing the exploitation of richer context, iv) the semantic correlation between column header and cell entity's description is exploited via a BERT-based cross-encoder.

Furthermore, DAGOBAH SL is packaged in a RESTful API [2] and user-friendly Web UI [13] to facilitate the usage of STI framework.

3. Lessons Learned and Challenges

Tabular data is highly heterogeneous. Relational table, as shown in Figure 1, is not the sole type of table. There are other types with different topology of the semantic connection between the cells, such as entity table[1] or matrix table[2]. In view of layout structure, tables can have multiple headers, splitted cells, merged cells or cells containing multiple values (e.g. a list). A more detailed table taxonomy is provided in [1].

Arguably, an annotation algorithm designed for a specific table type or table layout may not be suitable nor efficient for other types or layouts. DAGOBAH SL is built upon intuitions and assumptions derived from relational tables where i) each row corresponds to the description of a specific entity with columns providing its attributes; ii) a semantic cell (i.e. a cell that contains a mention that can be disambiguated) is fully represented by an entity; iii) tables have either no header or only a single header. However, the former intuition does not hold for matrix table and the two later assumptions hinder DAGOBAH SL from handling splitted cells and multivalued cells.

Knowledge Base (KB) Indexing and Exploitation. As a *closed Information Extraction* application [14], STI systems rely on a knowledge base (e.g. a knowledge graph, an ontology or a catalog of entity definitions) to

[1]See Fig. 1b in webtables for an example of entity table
[2]See Fig. 1c in webtables for an example of matrix table

constrain and guide the annotation process. A typical KB consists of millions of entities, requiring a proper indexing strategy for efficient retrieval and exploitation. The usage of a KB in information extraction tasks may imply two aspects: i) entity attributes to be indexed and exploited: entity can be characterized by labels/aliases, description or contextual information within a graph. These attributes are leveraged partly or fully to retrieve and disambiguate relevant entity candidates, ii) skewness of entity/relation distribution: entities and relations vary significantly in term of popularity and expressiveness, which may challenge the consistency in model performance. Heuristic-based annotation system, like DAGOBAH SL, relying on inverted indexes associated with flexible matching mechanism (e.g. exact matching or fuzzy matching) can work effectively and efficiently on tables that exhibit a high degree of literal similarity with entity's attributes. On the other hand, representation learning-based models learn the underlying semantics of table and entity through embeddings, making them more robust to noise and ambiguous/incomplete context. However, their performance depends strongly on the quality of training data and can differ greatly between frequently occurring entities/relations and rare ones.

Error Propagation from Detection to Annotation. Like many other STI systems [15, 16, 17, 18], DAGOBAH SL performs target detection (via *preprocessing*) and target annotation independently, hence suffers from error accumulation where error caused by the first stage will propagate to the later stage. Cases in which the system fails to distinguish cells containing literal mentions and cells containing semantic mentions, will lead to missing or incorrect annotations. Moreover, most target detection techniques are heuristic-based (e.g cells with string data type are considered as CEA targets), or locally contextualized (e.g. using only the single column to determine whether the inner cells are linkable to KG entities). The effectiveness of these techniques in various table scenarios remains uncertain.

Candidate Generation is challenging. Entity candidate generation is critical for effective STI systems that rely on the retrieve-then-rerank paradigm. Its goal is to deal with huge number of entities and narrow down the search space. DAGOBAH SL employs *dictionary lookup* that computes the literal similarity between the table mention (possibly with context) and entity's attributes (labels, aliases or descriptions). While this approach is appealing for handling various surface forms of mentions (e.g. acronym, synonym, typos), it lacks semantic understanding which can amplify the ambiguity within candidate sets and make the subsequent candidate re-ranking phase more challenging. With the rise of pre-trained language models, the embedding approach via contrastive learning based on dual encoders is capable of capturing both table and entity semantics, hence, can offer superior disambiguation capability. However, to the best of our knowledge, there has been no work on investigating the potential of dual encoders specifically on structured tabular data. Moreover, the construction of negative sets for contrastive learning is a non trivial task that impacts the quality of learned embeddings. In addition, likewise the target detection module, the candidate generation exhibits the risk of propagating errors to the annotation step. Hence, if the gold entity is not part of the retrieved candidates, the corresponding table element will never be correctly annotated.

4. Towards Generative Semantic Table Interpretation

Pre-trained generative LLMs (or foundational language models) have revolutionized numerous natural language understanding tasks, including information extraction (IE) tasks such as named entity recognition, entity linking, relation extraction [19]. The present state-of-the-art IE models leverage the flexibility of decoder-only or encoder-decoder LLM architectures for structured prediction, allowing for the joint handling of different IE tasks in an end-to-end and unified manner [20, 21, 22, 23, 24]. While this approach is originally applied to unstructured textual data, we argue it is still beneficial for structured data. In line with [25, 26], we believe that LLMs will be more and more adopted to tackle this task. This paper introduces our vision towards generative close Information Extraction tailored for tabular data, namely Generative Semantic Table Interpretation (GenSTI, Figure 2). As a stepping stone, we will aim to evaluate the potential of LLMs in tackling challenges discussed in Section 3. The desiderata are as follows:

Ability to handle simultaneously various table types and layouts. By framing the STI tasks within a unified seq-2-seq framework [27], generative LLMs can be prompted with different table types/layouts and can jointly solve CEA, CTA, CPA tasks. This framework reveals the common multi-task learning that has been successful in NLP. In particular, the model is fine-tuned with a mix of table sets to perform multiple STI tasks at once. Accordingly, it can facilitate knowledge transfer between tasks and table structures, leading to the acquisition of more robust and generalizable representations. Inspired by generative information extractors for text, we investigate two sequence modeling strategies: (i) *serialize table input and annotation outputs as plain text (text-2-text)* and solve with natural language LLMs (NL-LLMs) (Figure 2-left) [20, 21, 23]. This flattening method ignores,

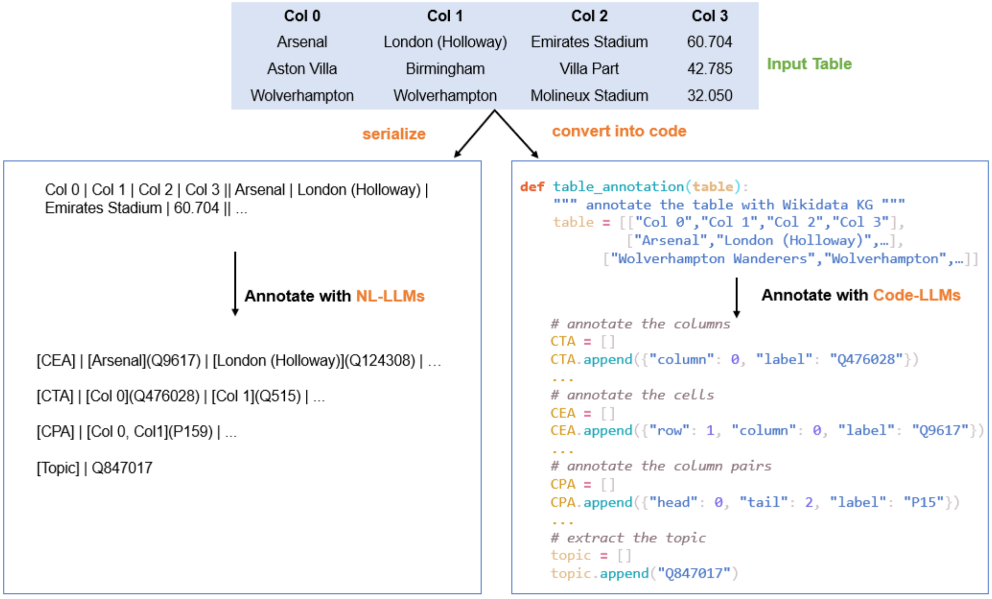

Figure 2: Generative Semantic Table Annotation with natural language-LLMs (left) and Code-LLMs (right).

however, the structural information embedded in table and output, due to the discrepancy between the datasets used to finetune LLMs for STI tasks and the natural language corpora that LLMs are pretrained on. To alleviate this issue, we could rely on [24, 28] to (ii) *cast STI tasks as code generation (code-2-code)* and solve with Code-LLMs (Figure 2-right). Converting structured data into code is easier and provides more informative representation than transforming it into free-form text. Hence, this technique narrows the gap between pre-training and fine-tuning in Code-LLMs. Interestingly, by programming a table as a two-dimensional list which is a common data type in code and is expected to occur frequently during the pre-training, the model could better capture the table's topology (i.e. facilitate the identification of the m_{th} row, n_{th} column or the cell at coordinates $[m, n]$), compared to NL-LLMs. [24, 28] have demonstrated the appealing few-shot performance of Code-LLMs (i.e. Codex[29]) in structured prediction task that involves no code at all such as information extraction or argument graph generation. We argue that using Code-LLMs to tackle table-related downstream tasks could be a promising future research direction.

End-to-End Semantic Table Annotation. Instead of performing the target detection and target annotation as separate stages, end-to-end STI takes into account the mutual dependence and cooperation between the two, which could lead to significant performance improvement. This concept has successfully inspired state-of-the-art models in related applications such as entity linking [22, 30] or information extraction [20, 23]. In the context of GenSTI, there are two aspects related to arbitrary generation of generative LLMs that need to be controlled to ensure a reliable end-to-end GenSTI: (i) *Output structure consistency*: only generate valid tags and adhere to the predefined schema. For example, in Figure 2, regarding CEA task, NL-LLM has to follow the template: [CEA] | cell_mention (entity_id) | ... where cell_mention is copied from the table and is linked to the corresponding entity_id in the KB; (ii) *Semantic consistency*: entity_id must be a valid entity existing in the KB, [CPA] must generate a relation_id rather than an entity_id. A solution to both challenges is to endow LLMs with a decoding scheme constrained by prefix-trees that forces the model to generate only legal tokens at each decoding step [22, 23]. Interestingly, even without such constraint decoding, we observe that [24] still reports good few-shot performance for IE tasks, suggesting that, to some extent, LLMs, especially Code-LLMs, are capable of capturing the internal representations of the task, and generate relevant outputs without guidelines. [31] made a similar observation when training a LLM to play Othello game by feeding it a naive transcript recording interleaving moves of two players without adding any knowledge of the game rules. The model has effectively learned meaningful latent representations, enabling it to uncover the game and make legal disc moves on the board.

Efficient KB indexing and exploitation. Recently, *Differentiable Search Index* (DSI) [32, 33] has emerged as a novel generative retrieval, deviating from the common retrieve-then-rerank paradigm. One of our objectives is to investigate whether DSI can serve as a viable solution for efficient KB indexing and exploitation within the GenSTI system. Specifically, entities and its graphs (e.g. [34]) are directly encoded (or stored) into LLM's parameters (aka. *indexing*). The model then leverages injected knowledge to predict autoregressively the entity identifier through softmax calculations over vocabulary's tokens. This indexing mechanism helps to relax the **Candidate Generation** phase which consequently saves a non-negligible computation cost and eliminate the need of external space to store entity embeddings and the need of meaningful negative samples for the learning, as required by dual encoders-based candidate generation. While working effectively on small KB, the behavior of DSI when scaling to large KB (e.g. Wikidata with ~ 100 millions entities) remains an open research challenge [35], necessitating three key elements to be clarified: (i) in the *indexing* phase, how many entities the model can memorize [36]; (ii) can the model efficiently learn entities and propagate its knowledge to support the generation [37]; and (iii) the robustness to entity/relation skewness. Generative Information Extraction [23] has shown to be more robust than strong baselines for long-tail entities, but is still far from being good. [38] proposes fine-tuning the extractor on a more balanced dataset that remarkably improves the macro performance.

5. Conclusion

This paper has first reflected on the work carried out when designing DAGOBAH-SL over the last few years and the lessons we learned in implementing a top performing system for STI. This work is also the first step in our journey towards Generative Semantic Table Interpretation (GenSTI). We plan to conduct intensive experiments to uncover the challenges and gain a deeper understanding of the contributions of LLMs to the STI topic, as discussed in Sections 3 and 4.

References

[1] J. Liu, Y. Chabot, R. Troncy, V.-P. Huynh, T. Labbé, P. Monnin, From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods, Journal of Web Semantics 76 (2023).

[2] Y. Chabot, F. Deuzé, V.-P. Huynh, T. Labbé, J. Liu, P. Monnin, R. Troncy, A Framework for Automatically Interpreting Tabular Data at Orange, in: ISWC (Posters/Demos/Industry Track), 2021.

[3] V. Cutrona, F. Bianchi, E. Jiménez-Ruiz, M. Palmonari, Tough tables: Carefully evaluating entity linking for tabular data, in: 9th International Semantic Web Conference (ISWC), Springer, 2020, pp. 328–343.

[4] E. Jiménez-Ruiz, O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, SemTab 2019: Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems, in: European Semantic Web Conference (ESWC), 2020.

[5] E. Jiménez-Ruiz, O. Hassanzadeh, V. Efthymiou, J. Chen, K. Srinivas, V. Cutrona, Results of SemTab 2020, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, 2020.

[6] V. Cutrona, J. Chen, V. Efthymiou, O. Hassanzadeh, E. Jiménez-Ruiz, J. Sequeda, K. Srinivas, N. Abdelmageed, M. Hulsebos, D. Oliveira, et al., Results of SemTab 2021, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, 2021.

[7] N. Abdelmageed, J. Chen, V. Cutrona, V. Efthymiou, O. Hassanzadeh, M. Hulsebos, E. Jiménez-Ruiz, J. Sequeda, K. Srinivas, Results of SemTab 2022, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching, 2022.

[8] M. Hulsebos, S. Gathani, J. Gale, I. Dillig, P. Groth, Ç. Demiralp, Making Table Understanding Work in Practice, arXiv preprint arXiv:2109.05173, 2021.

[9] V.-P. Huynh, Y. Chabot, T. Labbé, J. Liu, R. Troncy, From Heuristics to Language Models: A Journey Through the Universe of Semantic Table Interpretation with DAGOBAH, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), 2022.

[10] V.-P. Huynh, J. Liu, Y. Chabot, F. Deuzé, T. Labbé, P. Monnin, R. Troncy, DAGOBAH: Table and Graph Contexts for Efficient Semantic Annotation of Tabular Data, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), 2021.

[11] M. Hulsebos, Ç. Demiralp, P. Groth, Gittables: A large-scale corpus of relational tables, Proceedings of the ACM on Management of Data 1 (2023) 1–17.

[12] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, W. Fedus, Emergent Abilities of Large Language Models, Transactions on Machine Learning Research (2022).

[13] C. Sarthou-Camy, G. Jourdain, Y. Chabot, P. Monnin, F. Deuzé, V.-P. Huynh, J. Liu, T. Labbé, R. Troncy, DAGOBAH UI: a new hope for semantic table interpretation, in: European Semantic Web Conference (ESWC), Satellite Events, Springer, 2022, pp. 107–111.

[14] J. L. Martinez-Rodriguez, A. Hogan, I. Lopez-

Arevalo, Information extraction meets the semantic web: a survey, Semantic Web 11 (2020) 255–335.
[15] M. Cremaschi, F. De Paoli, A. Rula, B. Spahiu, A fully automated approach to a complete semantic table interpretation, Future Generation Computer Systems 112 (2020) 478–500.
[16] P. Nguyen, I. Yamada, N. Kertkeidkachorn, R. Ichise, H. Takeda, Demonstration of MTab: Tabular Data Annotation with Knowledge Graphs, in: ISWC (Posters/Demos/Industry Track), 2021.
[17] R. Shigapov, P. Zumstein, J. Kamlah, L. Oberländer, J. Mechnich, I. Schumm, bbw: Matching csv to wikidata via meta-lookup, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), 2020.
[18] N. Abdelmageed, S. Schindler, JenTab: Matching Tabular Data to Knowledge Graphs, in: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab), 2020, pp. 40–49.
[19] H. Ye, N. Zhang, H. Chen, H. Chen, Generative knowledge graph construction: A review, in: International Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2022.
[20] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, H. Wu, Unified Structure Generation for Universal Information Extraction, in: 60th Annual Meeting of the Association for Computational Linguistics (ACL), 2022, pp. 5755–5772.
[21] G. Paolini, B. Athiwaratkun, J. Krone, J. Ma, A. Achille, R. Anubhai, C. N. dos Santos, B. Xiang, S. Soatto, Structured Prediction as Translation between Augmented Natural Languages, in: 9th International Conference on Learning Representations (ICLR), 2021.
[22] N. De Cao, G. Izacard, S. Riedel, F. Petroni, Autoregressive entity retrieval, in: 9th International Conference on Learning Representations (ICLR), 2021.
[23] M. Josifoski, N. De Cao, M. Peyrard, F. Petroni, R. West, GenIE: Generative information extraction, in: Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Association for Computational Linguistics, Seattle, United States, 2022.
[24] P. Li, T. Sun, Q. Tang, H. Yan, Y. Wu, X. Huang, X. Qiu, CodeIE: Large Code Generation Models are Better Few-Shot Information Extractors, in: 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023.
[25] N. Tang, J. Fan, F. Li, J. Tu, X. Du, G. Li, S. Madden, M. Ouzzani, RPT: relational pre-trained transformer is almost all you need towards democratizing data preparation, The VLDB Endowment (2021).
[26] W.-C. Tan, Unstructured and structured data: Can we have the best of both worlds with large language models?, arXiv preprint arXiv:2304.13010, 2023.
[27] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 5485–5551.
[28] A. Madaan, S. Zhou, U. Alon, Y. Yang, G. Neubig, Language Models of Code are Few-Shot Commonsense Learners, in: International Conference on Empirical Methods in Natural Language Processing (EMNLP), Abu Dhabi, UAE, 2022.
[29] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al., Evaluating large language models trained on code, arXiv preprint arXiv:2107.03374 (2021).
[30] N. Kolitsas, O.-E. Ganea, T. Hofmann, End-to-End Neural Entity Linking, in: 22nd Conference on Computational Natural Language Learning (CoNLL), 2018, pp. 519–529.
[31] K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, M. Wattenberg, Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task, in: 11th International Conference on Learning Representations (ICLR), 2023.
[32] Y. Tay, V. Tran, M. Dehghani, J. Ni, D. Bahri, H. Mehta, Z. Qin, K. Hui, Z. Zhao, J. Gupta, et al., Transformer memory as a differentiable search index, Advances in Neural Information Processing Systems 35 (2022) 21831–21843.
[33] M. Bevilacqua, G. Ottaviano, P. Lewis, S. Yih, S. Riedel, F. Petroni, Autoregressive search engines: Generating substrings as document identifiers, Advances in Neural Information Processing Systems 35 (2022) 31668–31683.
[34] F. Moiseev, Z. Dong, E. Alfonseca, M. Jaggi, SKILL: Structured Knowledge Infusion for Large Language Models, in: Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2022, pp. 1581–1588.
[35] R. Pradeep, K. Hui, J. Gupta, A. D. Lelkes, H. Zhuang, J. Lin, D. Metzler, V. Q. Tran, How Does Generative Retrieval Scale to Millions of Passages?, in: Generative Information Retrieval @ SIGIR, 2023.
[36] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, C. Zhang, Quantifying Memorization Across Neural Language Models, in: 11Th International Conference on Learning Representations (ICLR), 2023.
[37] Y. Onoe, M. J. Zhang, S. Padmanabhan, G. Durrett, E. Choi, Can LMs Learn New Entities from Descriptions? Challenges in Propagating Injected Knowledge, in: 61st Annual Meeting of the Association for Computational Linguistics (ACL), 2023.
[38] M. Josifoski, M. Sakota, M. Peyrard, R. West, Exploit-

ing asymmetry for synthetic training data generation: Synthie and the case of information extraction, arXiv preprint arXiv:2303.04132, 2023.