FT-Mini: Overcoming the LLM Scale-Cost Trade-off in End-to-End Knowledge Graph Construction*

Apostolos Baniotis[1,2][0009−0006−3122−1844], Iordanis Sapidis[1,2][0009−0005−1367−7015], Valantis Zervos[1,2][0009−0002−9553−1535], Nikos Kontonasios[1,2][0009−0003−5929−0305], Katerina Papantoniou[1][0000−0003−1731−2951], Michalis Mountantonakis[1,2][0000−0002−1951−0241], Pavlos Fafalios[1,3][0000−0003−2788−526X], and Yannis Tzitzikas[1,2][0000−0001−8847−2130]

[1] Information Systems Laboratory (ISL), Institute of Computer Science (ICS), Foundation for Research and Technology - Hellas (FORTH), Greece
[2] Computer Science Department, University of Crete, Greece
[3] School of Production Engineering and Management, Technical University of Crete, Greece.
{baniotis,sapidis,vzervos,nkonto,papanton,mountant,fafalios,tzitzik}@ics.forth.gr

Abstract. Ontology-aware triple extraction from text requires identifying facts, linking entities, selecting valid properties, and producing consistent RDF representations. Large language models (LLMs) can support these tasks, but they are costly to run at scale, and multi-stage approaches propagate errors through intermediate representations. We present FT-Mini, a fine-tuned version of `gpt-4o-mini` that maps text directly to triples aligned to the DBpedia ontology. The model is trained on 100 examples and evaluated on the WebNLG family of benchmarks and on HotMusValid, a new diagnostic set for denser extraction settings that we introduce in this work. FT-Mini improves upon its base model and achieves F1 scores comparable to or higher than the evaluated GPT-5 configurations, with generally higher precision but lower recall. Inference through the batch API costs 95% less than GPT-5. The results suggest that targeted fine-tuning can improve extraction quality and reduce inference cost while maintaining conformity with the expected triple representations.

Keywords: knowledge graph construction · triple extraction · large language models · fine-tuning

1 Introduction

* This is a preprint of the paper: FT-Mini: Overcoming the LLM Scale-Cost Trade-off in End-to-End Knowledge Graph Construction, Apostolos Baniotis, Iordanis Sapidis, Valantis Zervos, Nikos Kontonasios, Katerina Papantoniou, Michalis Mountantonakis, Pavlos Fafalios and Yannis Tzitzikas, MTSR 2026, Bucharest, Romania

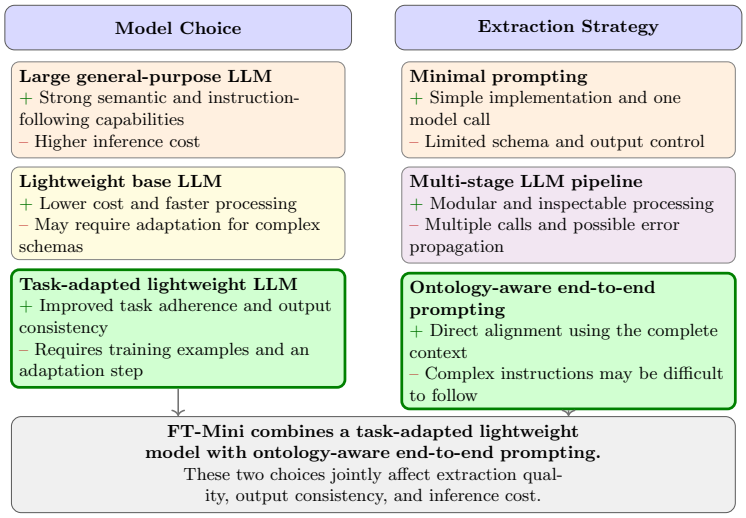


Fig. 1. General design choices for LLM-based ontology-aware triple extraction and their typical advantages and limitations. The highlighted options correspond to the configuration adopted by FT-Mini.

Knowledge Graphs (KGs) support semantic search, question answering, recommendation, and fact verification. Constructing KGs from unstructured text requires extracting entities and relations as schema-aligned triples [11,14]. Despite advances in Large Language Models (LLMs), generating source-grounded, ontology-compliant triples remains challenging [27,2]. This requires fact extraction, entity resolution, ontology-property selection, literal normalization, and consistent output generation.

Existing approaches face a cost–reliability trade-off. Task-specific models achieve high accuracy on fixed schemas but require substantial training data and a separate model per schema [24,16]. Prompted LLMs avoid this training cost and offer schema flexibility, but may generate unsupported facts, invalid properties, or inconsistent representations. Repeated use can also be costly for large document collections.

LLM-based extraction can perform extraction and ontology alignment through multiple stages or end-to-end. Multi-stage approaches can make operations easier to inspect, but intermediate errors or omissions may affect later stages. End-to-end approaches provide the full source context during entity and property selection, but lightweight models may follow complex instructions less reliably than larger models. Figure 1 summarizes these two design dimensions, model choice and extraction strategy, and their advantages and limitations.

We present FT-Mini, a `gpt-4o-mini` fine-tuned on 100 WebNLG and HotMusValid examples for end-to-end, DBpedia-aligned triple extraction. We evaluate it against minimal prompting, a three-stage GPT-5 pipeline, and end-to-end `gpt-4o-mini` and GPT-5 on WebNLG, WebNLG+, WebNLG++, and held-out HotMusValid texts. FT-Mini consistently outperforms its base model across these datasets, achieving F1 comparable to or higher than the tested GPT-5

configurations. It favors precision and output consistency, while GPT-5 achieves higher recall. Its batch API inference cost is 95% lower than GPT-5's. The main contributions are the following:

• An empirical investigation of low-data adaptation for end-to-end, DBpedia-aligned triple extraction using `gpt-4o-mini` fine-tuned on 100 selected examples.

• A comparison of fine-tuned and prompted extraction configurations, including a three-stage GPT-5 pipeline, showing F1 comparable to end-to-end GPT-5 at 95% lower batch API inference cost under the evaluated conditions; and

• HotMusValid, a diagnostic set for triple extraction from information-dense passages requiring cross-sentence reasoning.

The repository containing the code, prompts, training data and evaluation resources is publicly available on our GitHub[4].

The rest of this paper is organized as follows: §2 discusses the related work, §3 presents the methodology and §4 introduces the experimental setup. Moreover, §5 describes the results of the experiments whereas a discussion is provided in §6. Finally, §7 concludes the paper and discusses future directions.

2 Related Work

Triple extraction has been addressed by rule-based, statistical, and neural methods. Early Open Information Extraction systems used syntactic patterns and hand-crafted procedures on POS-tagged and dependency-parsed text [30,6]. Neural approaches subsequently used sequence labelling, relation classification, or joint entity–relation prediction. Pretrained language models improved performance under fixed relation schemas [24,16], but may not generalize across datasets [20].

2.1 LLM-based Triple Extraction

Recent work studies LLM-based triple extraction through prompting and fine-tuning. Prompting requires little or no additional training, but depends on instructions, demonstrations, and schema information. Papaluca et al. [17] incorporate KG context into LLaMA-65B prompts, while Wadhwa et al. [26] study prompting under limited supervision. Zhang et al. [32] show that smaller models trained with augmented examples can match larger prompted models, establishing small-model adaptation for triple extraction as prior work. REBEL [10], GenIE [12], and SynthIE [13] use sequence generation, targeting open extraction (REBEL) or closed extraction over Wikidata (GenIE, SynthIE).

LLM-based extraction also supports retrieval systems such as HippoRAG [8] and GeAR [23], which evaluate downstream retrieval rather than extraction quality. HippoRAG reports substantially better intrinsic OpenIE extraction than the task-specific REBEL model, motivating our use of a fine-tuned general-purpose

[4] `https://github.com/baniotis/ft-mini`

LLM. Inference cost is rarely reported in extraction studies. Prior work addresses distillation and smaller models [9,21], cost-performance trade-offs [4], and Green AI [22]. FT-Mini reports both extraction quality and inference cost.

Building on prior fine-tuning approaches, FT-Mini investigates end-to-end, DBpedia-aligned triple extraction using only 100 training examples without data augmentation. Unlike Zhang et al. [32], who focus on augmented training data, we evaluate low-data adaptation against untuned and larger prompted models, comparing extraction quality and inference cost. Our contribution is therefore an empirical evaluation of this specific adaptation strategy rather than a new fine-tuning method.

2.2 Evaluation Resources

NYT [19], TACRED [31], and DocRED [29] evaluate relation prediction over predefined relation sets and not complete ontology-aligned RDF generation. WebNLG [7], WebNLG+ [3], and WebNLG++ [1][5] instead pair texts with DBpedia triples. However, because their texts verbalize predefined triple sets, they provide a controlled extraction setting. Larger resources such as T-REx [5] scale to millions of Wikidata triples but rely on distant supervision, producing noisier alignments than the WebNLG family. We use the WebNLG family and introduce HotMusValid to complement it with denser passages.

3 Methodology

Given an input text D and a target ontology $\mathcal{O}$ with a set of valid properties $\mathcal{P}_{\text{target}}$, the objective is to generate a set of supported RDF triples

$$\mathcal{G} = \{(s, p, o) \mid p \in \mathcal{P}_{\text{target}}\}.$$

Subjects and entity-valued objects are represented as resource identifiers, predicates as DBpedia ontology or property URIs, and other objects as literals.

We compare three prompting strategies with FT-Mini: minimal one-shot prompting, a multi-stage pipeline, and end-to-end ontology-aware prompting. The prompts are available in the anonymous repository linked in Section 1.

3.1 Extraction Strategies

Minimal one-shot prompting. A short instruction and one example request DBpedia triples from the input text. Limited guidance on entity linking, property selection, literal handling, and serialization makes this strategy an unconstrained baseline.

[5] `https://github.com/baniotis/webnlg_plus_plus_dataset`

Multi-stage prompting. Extraction is decomposed into 3 sequential model calls:

1. Convert the source text into open-schema subject–predicate–object statements;
2. Map these statements to DBpedia resources and properties, including type assignment and literal formatting; and
3. Provide the source text and candidate ontology-aligned triples to the model for verification and correction.

Intermediate operations are inspectable, but later stages depend on earlier representations and may not recover information lost during initial extraction.

End-to-end ontology-aware prompting. A single call uses detailed instructions specifying entity linking preferences, DBpedia property selection, RDF type assertions, literal handling, and Turtle-style output conventions. Unlike the multi-stage pipeline, it avoids open-schema intermediate representations, allowing entity and property selection to consider the complete source text. Effectiveness depends on consistent adherence to these combined instructions. The prompt uses existing `dbr:` identifiers for confidently linked entities. For unlinked subjects, it instructs the model to construct candidate `dbr:` identifiers from normalized surface forms. These are RDF-compatible but do not confirm the existence of the these DBpedia resources. Evaluation thus assesses identifier construction and output consistency without verifying every generated identifier's existence.

3.2 FT-Mini Training

FT-Mini is a fine-tuned version of `gpt-4o-mini-2024-07-18`. It uses the same end-to-end instruction, output format, and generation temperature (set to 0.0 for determinism) as the prompted `gpt-4o-mini` configuration, so that the two configurations differ only in the fine-tuning step. Each training instance contains the instruction, an input text, and the expected DBpedia-aligned triples.

The training corpus contains 100 examples: 70 from WebNLG and 30 from HotMusValid. The corpus size was kept small to minimise annotation cost. Both development partitions are disjoint from their respective evaluation sets, and we manually checked that none of the training examples appear in the test sets of the WebNLG family or of HotMusValid. Candidates were scored by an automated script on relational density, syntactic complexity, coreference and distractor entities, using dependency-parse proxies combined as a weighted sum. To maintain lexical diversity, texts were clustered (TF-IDF, k-means) and the highest-scoring texts were selected with a cap per cluster. The top-ranked examples were then manually inspected. WebNLG provides controlled text–triple mappings, while HotMusValid contributes longer passages with cross-sentence dependencies.

To select the training duration, the corpus was first split into 70 training examples (50 WebNLG, 20 HotMusValid) and 30 validation examples (20 WebNLG, 10 HotMusValid). The validation loss was inspected across runs, and three epochs were selected. The final model was then fine-tuned for three epochs on all 100 examples, as the corpus is too small to reserve a held-out validation partition. All evaluation is performed on the separate test sets (see §4).

4 Experimental Setup

We evaluate the extraction strategies on WebNLG, WebNLG+, WebNLG++, and HotMusValid using the models and metrics described below.

4.1 Evaluation Protocol

Generated triples are evaluated using precision, recall, and F1. We apply two matching protocols: Exact Match and a combined matching protocol (referred to as `sameAs` in the evaluation code). Matching is one-to-one and greedy: each reference triple and each predicted triple is matched at most once. Exact Match accepts a prediction when its normalized representation (after stripping XSD suffixes, removing language tags, standardizing dates, etc.) is identical to a reference triple. This metric does not separately assess RDF datatype or language-tag preservation. Predictions that fail Exact Match are then evaluated by a semantic matching stage.

For entity-valued subjects and objects, two representations match when they are identical after normalization, resolve to the same URI through DBpedia Spotlight [15], or are connected through DBpedia `owl:sameAs` assertions. When both entities match, predicate similarity is estimated using the `all-MiniLM-L6-v2` sentence-embedding model [18]. A triple is accepted when

$$\mathrm{sim}(p_{\mathrm{pred}}, p_{\mathrm{ref}}) \geq 0.5 \quad \vee \quad \mathrm{sim}(p_{\mathrm{pred}} \oplus o_{\mathrm{pred}}, p_{\mathrm{ref}} \oplus o_{\mathrm{ref}}) \geq 0.75,$$

where sim is cosine similarity and $\oplus$ denotes concatenation. The thresholds were chosen empirically; we did not use a dedicated held-out set for this choice, so evaluation examples may have informed it.

The combined protocol reduces penalties for lexical and identifier variation, but embedding similarity does not establish logical equivalence between ontology properties. Exact Match results are therefore also reported separately for WebNLG. The evaluation implementation is available in the same anonymous repository as the prompts.

4.2 Datasets

WebNLG family. WebNLG [7] provides sets of DBpedia triples paired with crowdsourced verbalizations. Although designed for data-to-text generation, it also supports the reverse task: recovering a predefined set of triples from text that expresses them. WebNLG+ [3] extends the dataset with linguistic and structural annotations and adds Russian data; we use the English portion. WebNLG++ [1] updates the DBpedia resources, predicates, and factual content to better match the current state of the knowledge graph, making it a more realistic benchmark for RDF triple extraction from text[6].

[6] `https://github.com/baniotis/webnlg_plus_plus_dataset`

HotMusValid. HotMusValid is introduced in this paper to evaluate extraction from information dense passages derived from MuSiQue [25] and HotpotQA [28]. We treat it as a diagnostic set (rather than exhaustive benchmark), as its reference set contains the most prominent triples. We generated an initial set of candidate triples with Gemini 2.5 Pro using a DBpedia-oriented prompt, then the texts were divided among the seven annotators. Each annotator checked every candidate triple against DBpedia, looking up its subject and verifying the predicate and object, and cases that could not be resolved individually were discussed jointly in group meetings. Candidates were labelled *true* when explicitly supported, *false* when incorrect or unsupported, *synthetic* when inferable but not explicit, and *ignore* when ambiguous or not reliably mappable. Annotators could also add triples that were supported by the text but *missing* from the candidate set.

Of the 130 texts, 30 were used for development and 100 for evaluation, with no overlap. The latter contains 1,060 assessed candidate triples: 722 labelled *true* (495 accepted candidates and 227 manually added missing triples) and 338 *false*. Automated evaluation uses the *true* and *missing* triples as its reference set. On average, the reference set contains 7.2 triples per passage, comparable to the densest WebNLG family instances (up to 7 triples per instance). Many triples also require resolving information across sentences (e.g., in a passage describing "Soul Temple Records", the fact that Ghostface Killah is signed to it is only stated two sentences later, via the anaphoric reference "the label". The 30 development texts were verified in the same way, by checking each triple against DBpedia, but no synthetic triples were created for them.

HotMusValid evaluates text-grounded assertions expressed using DBpedia-oriented entities and properties. HotMusValid does not require its triples to already exist in DBpedia, and we do not verify that the identifiers generated by the model are valid DBpedia resources.

4.3 Models and Configurations

GPT-5 is evaluated using minimal one-shot prompting, the multi-stage pipeline, and end-to-end ontology-aware prompting. The `gpt-4o-mini-2024-07-18` model is evaluated using end-to-end prompting and serves as the base model for FT-Mini. FT-Mini uses the same base model, instruction, and output format differing only in the fine-tuning step described in Section 3.2. This comparison allows us to examine the effect of fine-tuning between the two `gpt-4o-mini` configurations, while comparison with end-to-end GPT-5 examines performance under the same extraction strategy. We record inference costs for the WebNLG++ runs through the OpenAI batch API and correspond to the inputs, outputs, and prices applicable during our experiments.

5 Results and Analysis

5.1 Comparative Extraction Performance

Table 1. F1 scores under the combined matching protocol.

Configuration	WebNLG	WebNLG+	WebNLG++	HotMusValid
Minimal GPT-5	0.4671	0.5205	0.5405	0.2511
Pipeline GPT-5	0.6104	0.6030	0.6598	0.2628
End-to-end `gpt-4o-mini`	0.5857	0.5884	0.6261	0.2812
End-to-end GPT-5	0.6810	0.6854	0.7084	0.2901
FT-Mini	**0.7130**	0.6943	0.7102	**0.3236**

Table 2. Exact Match performance on WebNLG.

Configuration	Prec.	Rec.	F1
Minimal GPT-5	0.1708	0.2119	0.1892
Pipeline GPT-5	0.3547	0.4506	0.3970
End-to-end `gpt-4o-mini`	0.3245	0.3192	0.3218
End-to-end GPT-5	0.4072	0.4719	0.4372
FT-Mini	**0.4947**	**0.5054**	**0.5000**

Table 1 reports F1 scores under the combined matching protocol. Minimal GPT-5 performs worst on the WebNLG datasets, while the multi-stage pipeline improves its scores but remains below end-to-end GPT-5 on all four datasets.

FT-Mini achieves the highest F1 across all datasets, outperforming its base model. Its advantage over end-to-end GPT-5 is larger on WebNLG (+0.032) and HotMusValid (+0.034), but marginal on WebNLG+ (+0.009) and WebNLG++ (+0.002); without repeated runs or significance testing, the latter differences should be interpreted as comparable performance. FT-Mini also achieves the highest precision across all datasets, while GPT-5 generally has higher recall. On WebNLG++, FT-Mini achieves 0.7237 precision and 0.6973 recall, compared with 0.6669 and 0.7555 for GPT-5, reflecting more conservative generation after fine-tuning. Absolute scores are lower on HotMusValid, reflecting its more complex passages and its non-exhaustive reference set. FT-Mini increases F1 from 0.2812 for its base model to 0.3236, but absolute scores remain low.

5.2 Exact-Match Evaluation

Table 2 reports Exact Match results for WebNLG. Scores are lower than under combined matching because lexical or identifier variations receive no credit. FT-Mini obtains the highest precision, recall, and F1, indicating that its predictions more often match the reference representations.

5.3 Exploratory Manual Audit

We manually examined predictions for ten randomly selected HotMusValid texts. We counted as correct any triple supported by the source text but absent from the reference set (which contains only the most prominent triples), as well as any conceptually correct assertion with serialization differences. Audit precision is the number of correct predictions divided by all predictions; audit recall is

Table 3. Automated and manual-audit scores on ten HotMusValid texts.

	`gpt-4o-mini`		FT-Mini	
Metric	**Auto**	**Audit**	**Auto**	**Audit**
Precision	0.2449	0.5612	0.2981	0.9420
Recall	0.2824	0.6627	0.3647	0.6740
F1	0.2623	0.6076	0.3280	0.7860

the number of original reference triples recovered through semantic matching divided by the size of the original reference set for these ten texts. The audit scores are not directly comparable to the automated results and should be read as a separate qualitative assessment.

Table 3 shows higher scores for both models under this assessment. For FT-Mini, several automated false positives were supported by the source text but absent from the reference set. The audit also identified actual extraction errors, including omitted relations, inappropriate entity representations, and incorrect literal normalization. Despite the small sample, these findings point to a broader issue: a triple may be semantically correct and grounded in the source text, yet be counted as a false positive because it is absent from a non-exhaustive reference set. Precision, recall, and F1 alone may therefore not fully capture extraction quality in this setting. This motivates evaluation methods that also assess textual support, semantic correctness, and reference completeness.

5.4 Inference Cost

Processing WebNLG++ (test set ([1,779] texts) through the batch API cost $0.98 with `gpt-4o-mini`, $1.12 with FT-Mini, and $23.15 with GPT-5 (default settings, temperature set to 0) using the prices in February 2026. FT-Mini reduced the inference cost by 95% relative to GPT-5 while achieving a comparable F1 score. These values are specific to the prompts, outputs, API prices, and batch configuration used in our experiments. The one-time fine-tuning cost was $1.14 for all fine-tuning runs, including the 70/30 diagnostic run.

6 Discussion

Fine-tuning improves conformity with reference representations, primarily through higher precision and Exact Match, while FT-Mini generally retains lower recall than GPT-5. This pattern suggests more conservative triple generation.

The multi-stage GPT-5 pipeline improves upon minimal prompting but remains below end-to-end GPT-5. Its intermediate open-schema triples often omit qualifying information or express relations too generally for unambiguous DBpedia alignment. The final stage receives the source text but does not always recover information lost in earlier stages. These findings concern the evaluated pipeline; other designs could preserve richer context or add iterative correction and explicit constraint checking.

FT-Mini's Exact Match results show increased conformity with reference representations, useful when triples require limited post-processing. However, exact correspondence does not establish factual correctness or RDF validity, and errors remain in literal normalization, entity representation, and relation coverage. For unlinked subjects, the prompt constructs candidate `dbr:` identifiers from their surface forms. Although RDF-compatible, these identifiers are not verified to exist in DBpedia and would need verification or replacement with locally controlled URIs in a deployed system.

The cost results demonstrate the potential value of task adaptation under the evaluated API configurations, but exclude annotation, fine-tuning, and downstream validation costs. Adapting to other ontologies would require revised namespaces, instructions, and schema-aligned examples, but transfer beyond DBpedia remains unverified.

WebNLG and its variants provide controlled text–triple mappings that do not fully reflect naturalistic extraction. The combined matching protocol may accept semantically related but ontologically distinct predicates, while Exact Match may reject valid lexical variations. HotMusValid has a non-exhaustive reference set, and its manual audit covers only ten texts. As the audit shows, text-supported, semantically correct triples absent from the reference may be counted as false positives. Ablations of training-data selection, dataset composition, and training duration, plus repeated runs and significance testing, remain future work. Because API-accessed models have undisclosed architectures and parameter counts, we scope our findings to the evaluated models, datasets, prompts, and service configurations.

7 Conclusion

This paper presented FT-Mini, a fine-tuned version of `gpt-4o-mini` for end-to-end, DBpedia-aligned triple extraction, and introduced HotMusValid, a diagnostic set for extraction from passages that require cross-sentence reasoning. Across four datasets, FT-Mini improves upon its base model and matches or exceeds GPT-5 in F1 at substantially lower inference cost. These results suggest that targeted fine-tuning can offset the scale advantage of larger models for well-defined extraction tasks, and that lightweight, task-adapted models remain a practical alternative to prompting large general-purpose LLMs.

Future work will examine larger human evaluations, transfer beyond DBpedia and to other languages, and fine-tuning of open-weight models.

Declaration on Generative AI The authors did not use generative AI systems for generating scientific content, analysis, results, figures or conclusions presented in the paper; Gemini 3.8 Flash was used for language refinement and proofreading.

References

1. Baniotis, A., Papantoniou, K., Mountantonakis, M., Kontonasios, N., Sapidis, I., Fafalios, P., Zervos, V., Tzitzikas, Y., Lai, R., Fejza, A., Piramanayagam, S., Vougiouklis, P., Pan, J.Z.: Adapting WebNLG to DBpedia Evolution for Evaluating LLM-based Triple Extraction. In: Workshop on Graph-enhanced LLMs for trustworthy Web data management (GLOW), in conjunction with ISWC'26. Bari, Italy (2026)
2. Bian, H.N.: LLM-empowered knowledge graph construction: A survey. ArXiv **abs/2510.20345** (2025), https://arxiv.org/abs/2510.20345
3. Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., Shimorina, A.: The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In: Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., Shimorina, A. (eds.) Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+). pp. 55–76. ACL, Dublin, Ireland (Virtual) (12 2020), https://aclanthology.org/2020.webnlg-1.7/
4. Chen, L., Zaharia, M., Zou, J.: FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. arXiv preprint arXiv:2305.05176 (2023)
5. Elsahar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., Simperl, E.: T-REx: A large scale alignment of natural language with knowledge base triples. In: Proceedings of the LREC 2018. ELRA, Miyazaki, Japan (May 2018)
6. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Conference on Empirical Methods in Natural Language Processing (2011), https://api.semanticscholar.org/CorpusID:10318045
7. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: The WebNLG challenge: Generating text from RDF data. In: Alonso, J.M., Bugarín, A., Reiter, E. (eds.) Proceedings of the 10th International Conference on Natural Language Generation. pp. 124–133. Association for Computational Linguistics, Santiago de Compostela, Spain (Sep 2017). https://doi.org/10.18653/v1/W17-3518, https://aclanthology.org/W17-3518/
8. Gutiérrez, B.J., Shu, Y., Gu, Y., Yasunaga, M., Su, Y.: HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. Advances in neural information processing systems **37**, 59532–59569 (2024)
9. Hsieh, C.Y., Li, C.L., Yeh, C.K., Nakhost, H., Fujii, Y., Ratner, A., Krishna, R., Lee, C.Y., Pfister, T.: Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. In: ACL Findings (2023)
10. Huguet Cabot, P.L., Navigli, R.: REBEL: Relation extraction by end-to-end language generation. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 2370–2381. ACL, Punta Cana, Dominican Republic (2021)
11. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems **33**(2), 494–514 (2022)
12. Josifoski, M., De Cao, N., Peyrard, M., Petroni, F., West, R.: GenIE: Generative information extraction. In: Proceedings of the NAACL 2022. pp. 4626–4643. ACL, Seattle, United States (2022)
13. Josifoski, M., Sakota, M., Peyrard, M., West, R.: Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In: Proceedings of the EMNLP 2023. ACL, Singapore (2023)

14. Martinez-Rodriguez, J.L., Hogan, A., Lopez-Arevalo, I.: Information extraction meets the semantic web: a survey. Semantic Web **11**(2), 255–335 (2020)
15. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: Shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8 (2011)
16. Ning, J., Yang, Z., Sun, Y., Wang, Z., Lin, H.: OD-RTE: A One-Stage Object Detection Framework for Relational Triple Extraction. In: Proceedings of the 61st Annual Meeting of the ACL. pp. 11120–11135 (2023)
17. Papaluca, A., Krefl, D., Rodríguez Méndez, S., Lensky, A., Suominen, H.: Zero- and few-shots knowledge graph triplet extraction with large language models. In: Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024). pp. 12–23. ACL, Bangkok, Thailand (Aug 2024)
18. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proceedings of the EMNLP 2019. pp. 3982–3992 (2019)
19. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Balcázar, J.L., Bonchi, F., Gionis, A., Sebag, M. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 148–163. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
20. Saini, P., Pal, S., Nayak, T., Bhattacharya, I.: 90% f1 score in relation triple extraction: Is it real? In: Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP. pp. 1–11. ACL, Singapore (Dec 2023)
21. Schick, T., Schütze, H.: It's not just size that matters: Small language models are also few-shot learners. In: Proceedings of the NAACL 2021 (2021)
22. Schwartz, R., Dodge, J., Smith, N.A., Etzioni, O.: Green ai. Communications of the ACM **63**(12), 54–63 (2020)
23. Shen, Z., Diao, C., Vougiouklis, P., Merita, P., Piramanayagam, S., Chen, E., Graux, D., Melo, A., Lai, R., Jiang, Z., Li, Z., QI, Y., Ren, Y., Tu, D., Pan, J.Z.: GeAR: Graph-enhanced Agent for Retrieval-augmented Generation. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 12049–12072 (2025)
24. Tang, W., Xu, B., Zhao, Y., Mao, Z., Liu, Y., Liao, Y., Xie, H.: Unirel: Unified representation and interaction for joint relational triple extraction. In: Proceedings of the EMNLP 2022. pp. 7087–7099 (2022)
25. Trivedi, H., Balasubramanian, N., Khot, T., Sabharwal, A.: MuSiQue: Multihop questions via single-hop question composition. Transactions of the Association for Computational Linguistics **10**, 539–554 (2022)
26. Wadhwa, S., Amir, S., Wallace, B.: Revisiting relation extraction in the era of large language models. In: Proceedings of the 61st Annual Meeting of the ACL (Volume 1: Long Papers). pp. 15566–15589. Association for Computational Linguistics, Toronto, Canada (Jul 2023)
27. Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., Wang, Y., Chen, E.: Large language models for generative information extraction: A survey (2024). https://doi.org/10.1007/s11704-024-40555-y
28. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering (2018), https://arxiv.org/abs/1809.09600
29. Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J., Sun, M.: DocRED: A large-scale document-level relation extraction dataset. In: Proceedings of the 57th Annual Meeting of the ACL. pp. 764–777. ACL, Florence, Italy (2019)

30. Yates, A., Banko, M., Broadhead, M., Cafarella, M., Etzioni, O., Soderland, S.: TextRunner: Open information extraction on the web. In: Carpenter, B., Stent, A., Williams, J.D. (eds.) Proceedings of Human Language Technologies: NAACL. pp. 25–26. ACL, Rochester, New York, USA (Apr 2007)
31. Zhang, Y., Zhong, V., Chen, D., Angeli, G., Manning, C.D.: Position-aware attention and supervised data improve slot filling. In: Proceedings of the EMNLP 2017. pp. 35–45. ACL, Copenhagen, Denmark (2017). https://doi.org/10.18653/v1/D17-1004
32. Zhang, Y., Sadler, T., Taesiri, M.R., Xu, W., Reformat, M.: Fine-tuning language models for triple extraction with data augmentation. In: Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024). pp. 116–124. ACL, Bangkok, Thailand (Aug 2024)