Wikontic: A Tool for Building Knowledge Graphs from Text Aligned with the Wikidata Ontology

Alla Chepurova[1,2], **Aydar Bulatov**[1,2], **Mikhail Burtsev**[3], **Yuri Kuratov**[1,2]
[1]Cognitive AI Systems Lab, Moscow, Russia
[2]Moscow Independent Research Institute of Artificial Intelligence, Moscow, Russia
[3]London Institute for Mathematical Sciences, London, UK
{chepurova, bulatov, kuratov}@cogailab.com, mb@lims.ac.uk

Abstract

Knowledge Graphs (KGs) provide structured, verifiable representations that ground facts and supply large language models (LLMs) with reliable real-world information. Building high-quality KGs from open-domain text remains difficult due to redundancy, inconsistency, and lack of ontology grounding. We present Wikontic, a pipeline that extracts triples from text with LLMs and refines them through ontology-based typing, schema validation, and entity deduplication, yielding compact and coherent graphs. Unlike prior frameworks that lack ontology grounding or perform only partial deduplication, Wikontic uniquely integrates entity canonicalization, alias tracking, and automatic enforcement of Wikidata's ontology, enabling robust schema-aware construction without manual schema design. Its web interface lets users upload text, visualize graphs, and perform multi-hop question answering. By combining LLM flexibility with Wikidata's ontological rigor, Wikontic transforms ambiguous text into structured, interpretable, and actionable knowledge.

Video — https://youtu.be/Aw0F3TJyGfQ
Demo — https://wikontic.deeppavlov.ai/
Code — https://github.com/screemix/Wikontic

Introduction and Related Work

A large portion of the world's knowledge is available only in unstructured text, from news and scientific articles to blogs and social media. Large language models (LLMs) can extract insights from such data, but their internal representations are latent and unverifiable, which makes them prone to hallucinations (Ji et al. 2023; Huang et al. 2025). Knowledge graphs (KGs) provide a structured alternative: subject–relation–object triples enable transparent reasoning, verifiable queries, and straightforward updates. As a result, KGs are reliable complements to LLMs and retrieval-augmented generation (RAG) systems serving as long-term reusable representations of textual knowledge.

The task of constructing KGs from text has long been studied in information retrieval and information extraction. Closed information extraction (cIE) methods enforce structure but are limited to predefined entity and relation sets from a static KG and domain-specific training data (Distiawan et al. 2019; Cabot and Navigli 2021; Josifoski et al. 2021, 2023; Chepurova et al. 2024; Guo et al. 2023). Recently, LLMs have emerged as powerful alternatives for open information extraction (oIE), where KGs are built from scratch without predefined entities, relations, and constraints. This setting offers flexibility and has been explored as an enhancement to RAG systems, where lightweight graph structures improve retrieval efficiency and answer reliability (Jimenez Gutierrez et al. 2024; Guo et al. 2025; Li et al. 2024; Han et al. 2024; Gutiérrez et al. 2025; Anokhin et al. 2024). However, oIE pipelines face a fundamental limitation: the lack of structure often results in noisy, redundant, and ambiguous graphs. Surface form variation (e.g., "NYC" vs. "New York City"), synonymy, and overlapping predicates fragment the graph, while the absence of schema constraints allows contradictory or incoherent triples. This undermines the very advantages of KGs: clarity, coherence, and logical precision.

Wikontic: System Overview

We address this gap with **Wikontic**, a pipeline that constructs KGs directly from raw text by combining LLM-based extraction, entity deduplication, and Wikidata's ontology constraints. This approach retains the flexibility of oIE while enforcing the structure, type hierarchies, and relational constraints of a large-scale curated ontology. Wikidata (Vrandečić 2012) is particularly suitable: it contains over 100M entities across domains, with rich typing, relation schemas, and community-maintained quality control. Its breadth allows coverage from common sense to specialized domains, while its formal constraints provide principled supervision for validating LLM outputs. Wikontic integrates six components: (i) a curated Ontology database derived from Wikidata, (ii) Candidate Triplet Extraction with qualifiers, (iii) Ontology-aware Triplet Refinement enforcing schema rules, (iv) Subject/Object Name Refinement for deduplication and alias tracking, (v) KG Storage with user-specific subgraphs, and (vi) Retrieval for multi-hop QA.

Ontology Database. To enable ontology-guided refinement, we curated an *Ontology database* from Wikidata containing 2,414 unique types of relations (excluding multimedia/external-link properties), their corresponding subject–object constraints and entity types. For each re-

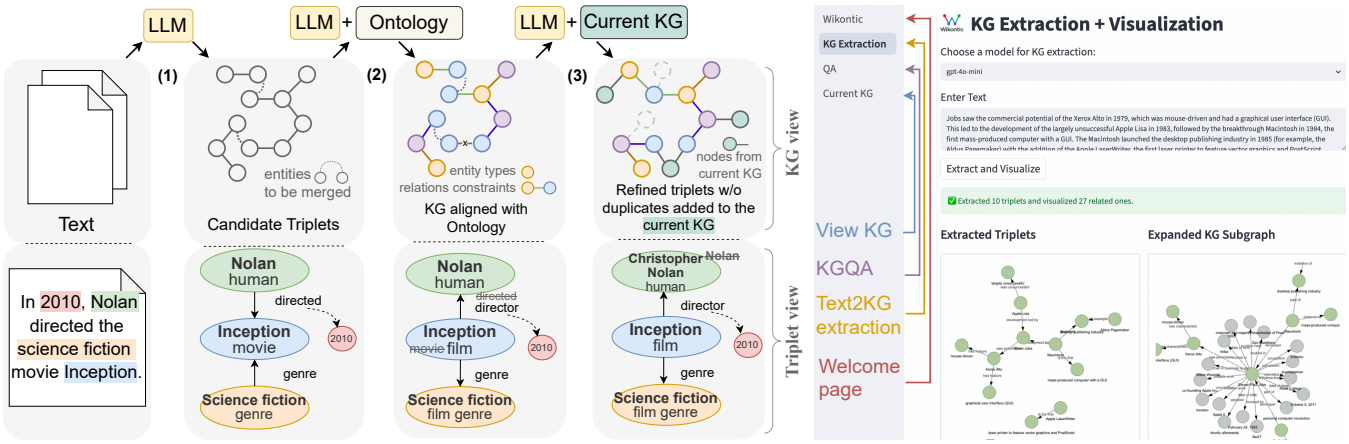

Figure 1: Overview of **Wikontic**'s pipeline (left): (1) candidate triplet extraction, (2) ontology-aware refinement, (3) entity name refinement; and overview of the user interface (right).

lation and entity type we gathered labels and aliases, then recursively expanded entity type hierarchies using *instance of* (P31) and *subclass of* (P279). Two dense retrieval indices, one for relation names and the other for entity-type names, map the surface forms produced by the LLM to the closest ontology entries, enabling schema-compliant refinement.

Candidate Triplet Extraction. We use LLMs to extract subject–relation–object triples enriched with subject/object types and contextual qualifiers (e.g., time, location, condition). Such output preserves factual content while capturing nuances that cannot be represented as standalone triples.

Ontology-aware Triplet Backbone Refinement. To ensure that each triplet extracted by the LLM represents a semantically meaningful and structurally valid fact, Wikontic performs an ontology-based refinement using the schema and constraints defined in Wikidata. Extracted subject/object entity types from each triplet produced in the first step are mapped to candidate types retrieved via vector search over Wikidata's labels and aliases, and expanded with their supertypes. Legal relations between candidate types are drawn from Wikidata's schema and ranked by embedding similarity to the original relation. The LLM then rewrites the triple with ontology-compliant type-relation pair, producing facts that are both semantically aligned and structurally valid.

Subject and Object Name Refinement. To avoid duplication and ensure consistency within the constructed KG, each entity mention is linked to existing same-type nodes. Candidates are retrieved by embedding similarity and then verified by the LLM. Approved matches adopt the canonical label from KG while the original mention is stored as an alias. Unmatched mentions become new nodes. This keeps the KG compact yet evolving, enabling both entity reuse and new concept discovery.

KG storage and Retrieval. The *KG database* stores extracted triples, entity names, and aliases. Dense retrieval index over entity aliases supports efficient deduplication and entity linking. For multi-hop QA, relevant entities are first extracted from the question using an LLM, linked to KG entries, and expanded into their 5-hop neighborhoods. These triples and their qualifiers provide a structured, grounded context that the LLM can use to generate accurate answers.

Method	Deduplication	Ontology
LlamaIndex (LlamaIndex 2023)	No	No
LangChain (LangChain 2022)	No	Partial
Neo4j (Neo4j Labs 2024)	Partial	Partial
KGGen (Mo et al. 2025)	Yes	No
GraphRAG (Edge et al. 2024)	Partial	No
HippoRAG (Jimenez Gutierrez et al. 2024)	Partial	No
AriGraph (Anokhin et al. 2024)	No	No
Wikontic (Ours)	Yes	Wikidata

Table 1: Wikontic uniquely combines entity canonicalization and alias tracking with automatic Wikidata ontology enforcement, enabling robust, schema-aware KG construction.

User interface. A Streamlit web application allows users to upload text, select an LLM, visualize KGs, and ask questions grounded in the constructed KG. Each session maintains a user-specific subgraph, which can be cleared or extended on demand.

Conclusion and Future Work

Prior text-to-KG frameworks supporting retrieval for downstream tasks like QA either lack ontology grounding or offer only partial deduplication (Table 1), yielding incoherent or redundant graphs. Wikontic uniquely combines explicit entity canonicalization and alias tracking with automatic Wikidata's ontology enforcement, enabling robust, schema-aware KG construction without manual schema design.

By uniting expressive LLM extraction with Wikidata's ontological rigor, Wikontic produces compact, verifiable KGs suited for reasoning, retrieval, and LLM grounding. Unlike prior frameworks Wikontic uniquely integrates entity canonicalization, alias tracking, and automatic ontology enforcement for schema-aware construction without manual design. Our demonstration shows how LLM-driven extraction can be transparent, verifiable, and reusable, bridging unstructured language and structured reasoning. Future work will extend these capabilities to additional ontologies and domains, broadening the scope of transparent, verifiable knowledge integration.

Acknowledgements

A.C., A.B., and Y.K.'s work was supported by the Ministry of Economic Development of the Russian Federation (Agreement No. 139-15-2025-013, dated June 20, 2025, IGK 000000C313925P4B0002).

References

Anokhin, P.; Semenov, N.; Sorokin, A.; Evseev, D.; Burtsev, M.; and Burnaev, E. 2024. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. *arXiv preprint arXiv:2407.04363*.

Cabot, P.-L. H.; and Navigli, R. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2370–2381.

Chepurova, A.; Kuratov, Y.; Bulatov, A.; and Burtsev, M. 2024. Prompt Me One More Time: A Two-Step Knowledge Extraction Pipeline with Ontology-Based Verification. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, 61–77.

Distiawan, B.; Weikum, G.; Qi, J.; and Zhang, R. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 229–240.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Guo, K.; Diefenbach, D.; Gourru, A.; and Gravier, C. 2023. Wikidata as a seed for Web Extraction. In *Proceedings of the ACM Web Conference 2023*, 2402–2411.

Guo, Z.; Xia, L.; Yu, Y.; Ao, T.; and Huang, C. 2025. LightRAG: Simple and Fast Retrieval-Augmented Generation. arXiv:2410.05779.

Gutiérrez, B. J.; Shu, Y.; Qi, W.; Zhou, S.; and Su, Y. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Han, H.; Wang, Y.; Shomer, H.; Guo, K.; Ding, J.; Lei, Y.; Halappanavar, M.; Rossi, R. A.; Mukherjee, S.; Tang, X.; et al. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2): 1–55.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12): 1–38.

Jimenez Gutierrez, B.; Shu, Y.; Gu, Y.; Yasunaga, M.; and Su, Y. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37: 59532–59569.

Josifoski, M.; De Cao, N.; Peyrard, M.; Petroni, F.; and West, R. 2021. GenIE: Generative information extraction. *arXiv preprint arXiv:2112.08340*.

Josifoski, M.; Sakota, M.; Peyrard, M.; and West, R. 2023. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. *arXiv preprint arXiv:2303.04132*.

LangChain. 2022. LangChain Documentation. https://docs.langchain.dev.

Li, S.; He, Y.; Guo, H.; Bu, X.; Bai, G.; Liu, J.; Liu, J.; Qu, X.; Li, Y.; Ouyang, W.; et al. 2024. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. *arXiv preprint arXiv:2406.14550*.

LlamaIndex. 2023. LlamaIndex Documentation. https://docs.llamaindex.ai.

Mo, B.; Yu, K.; Kazdan, J.; Mpala, P.; Yu, L.; Cundy, C.; Kanatsoulis, C.; and Koyejo, S. 2025. KGGen: Extracting Knowledge Graphs from Plain Text with Language Models. *arXiv preprint arXiv:2502.09956*.

Neo4j Labs. 2024. Neo4j LLM Knowledge Graph Builder. https://github.com/neo4j-labs/llm-graph-builder.

Vrandečić, D. 2012. Wikidata: a new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, 1063–1064. New York, NY, USA: Association for Computing Machinery.