




HeXtractor: A Tool for Building Heterogeneous Graphs from Structured and Textual Data for Graph Neural Networks

First Author¹ and Second Author¹

¹ Example Institution, Country

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright
and release the work under a
Creative Commons Attribution 4.0
International License ([CC BY 4.0](#)).

Summary

HeXtractor is an open-source Python library designed to transform structured tabular data and unstructured textual content into heterogeneous graph representations suitable for use in Graph Neural Networks (GNNs). Fully compatible with the PyTorch Geometric (PyG) framework (Fey & Lenssen, 2019), HeXtractor provides a streamlined, high-level interface for defining entities (nodes), relationships (edges), and associated metadata across diverse data modalities.

GNNs have become increasingly prominent with the rise of the Message Passing Neural Network (MPNN) paradigm (Gilmer et al., 2017). In particular, **heterogeneous graphs**, which support multiple node and edge types, are gaining traction in domains such as recommendation systems, fraud detection, and knowledge representation (Shi, 2022; Yang et al., 2020). Advanced architectures—including Heterogeneous Graph Transformers (Hu et al., 2020) and Heterogeneous Graph Attention Networks (X. Wang et al., 2019)—are specifically designed to leverage the rich semantics of these graph types.

A key application of heterogeneous graphs is in **knowledge graph construction**, which models complex, real-world relationships. Such graphs are used in a variety of industries, including job-market matching (Chen et al., 2018; Noy et al., 2019) and credit risk analysis (Mitra et al., 2024).

Despite their utility, constructing heterogeneous graphs remains a labor-intensive and error-prone task. HeXtractor addresses this challenge by offering a standardized, automated tool to convert structured and unstructured data into GNN-compatible formats, with optional support for large language models (LLMs) for extracting structure from text.

Statement of Need

A **heterogeneous graph** is formally defined as a tuple $G = (V, E)$, where V and E represent sets of nodes and edges, respectively. Each node $v \in V$ and edge $e \in E$ is associated with a type mapping: $\phi(v) : V \rightarrow A$ and $\Phi(e) : E \rightarrow R$, where A and R denote sets of node and edge types (Shi, 2022). These graphs capture both the structural and semantic heterogeneity inherent in many real-world datasets.

While libraries such as PyG (Fey & Lenssen, 2019) and DGL (M. Wang et al., 2019) provide robust learning capabilities for such graphs, they offer limited tooling for graph construction—particularly when data is distributed across multiple heterogeneous sources. As a result, researchers often rely on custom-built scripts, introducing variability and undermining reproducibility.

HeXtractor addresses this gap by providing:

- A declarative interface for defining node and edge schemas;
- Integration with LLMs for extracting graph structure from natural language using LangChain-compatible GraphDocument objects;
- Schema validation and consistency checks;
- Interactive graph visualization capabilities;
- Seamless export to PyG's HeteroData format.

Initially developed as part of the **HexGIN** project (Wójcik, 2024), which focused on analyzing financial transaction data, HeXtractor has evolved into a domain-agnostic tool for heterogeneous graph extraction.

Features and Usage

HeXtractor enables graph construction from both structured tabular datasets and unstructured textual content. It also supports visualization and full interoperability with the PyTorch Geometric framework.

Structured Data Extraction

HeXtractor supports both single-table and multi-table data processing. In single-table mode, each row encodes a relationship among column-defined entities. Users specify:

1. Node types and their attributes;
2. Edge definitions among the entities.

This yields a PyG-compatible HeteroData object ready for downstream modeling:

```
HeteroData(  
    company={ x=[3, 2] },  
    employee={ x=[7, 2], y=[7] },  
    tag={ x=[5] },  
    (company, has, employee)={ edge_index=[2, 6] },  
    (company, has, tag)={ edge_index=[2, 7] }  
)
```

Interactive visualization is supported, with customizable labels and color schemes to aid interpretability.

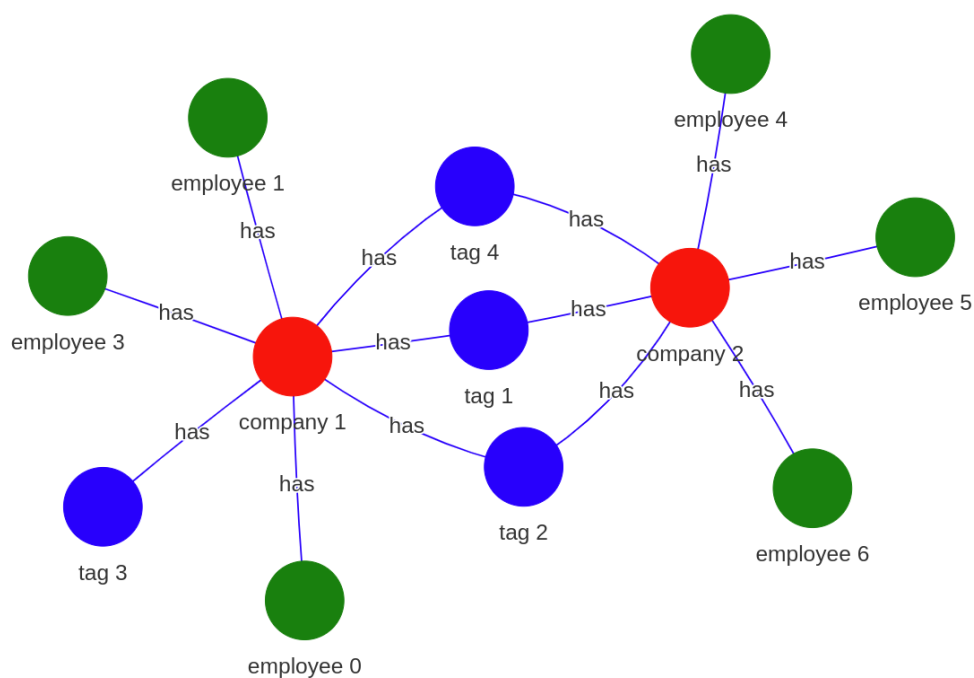


Figure 1: Graph extracted from structured data.

⁶⁰ In multi-table mode, HeXtractor utilizes user-defined GraphSpecs to combine entity and
⁶¹ relationship tables into a unified heterogeneous graph.

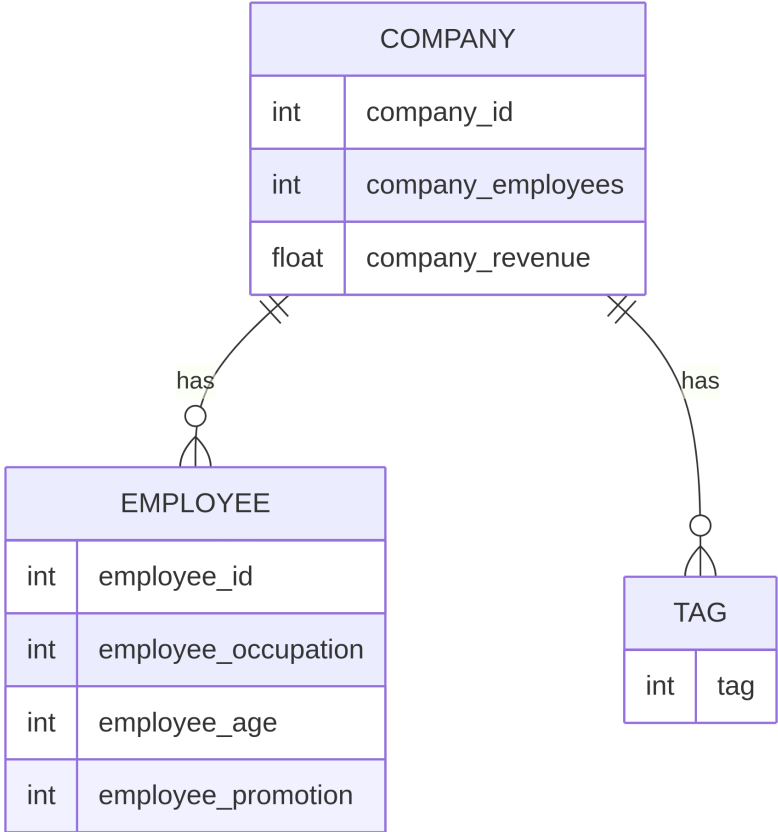


Figure 2: Entity relationship diagram.

62 **Text-Based Graph Extraction**

63 Through integration with **LangChain**, HeXtractor supports automated extraction of semantic
64 graph structures from natural language. The process is as follows:

- 65 1. Input text is processed by an LLM.
66 2. The model outputs a GraphDocument containing nodes and relationships.
67 3. HeXtractor converts the result into a HeteroData object.

68 For instance, the following input:

69 Marcin Malczewski and Filip Wójcik are data scientists
70 who developed HeXtractor. It helps extract heterogeneous
71 knowledge graphs from various data sources.

72 is transformed into the following heterogeneous graph:

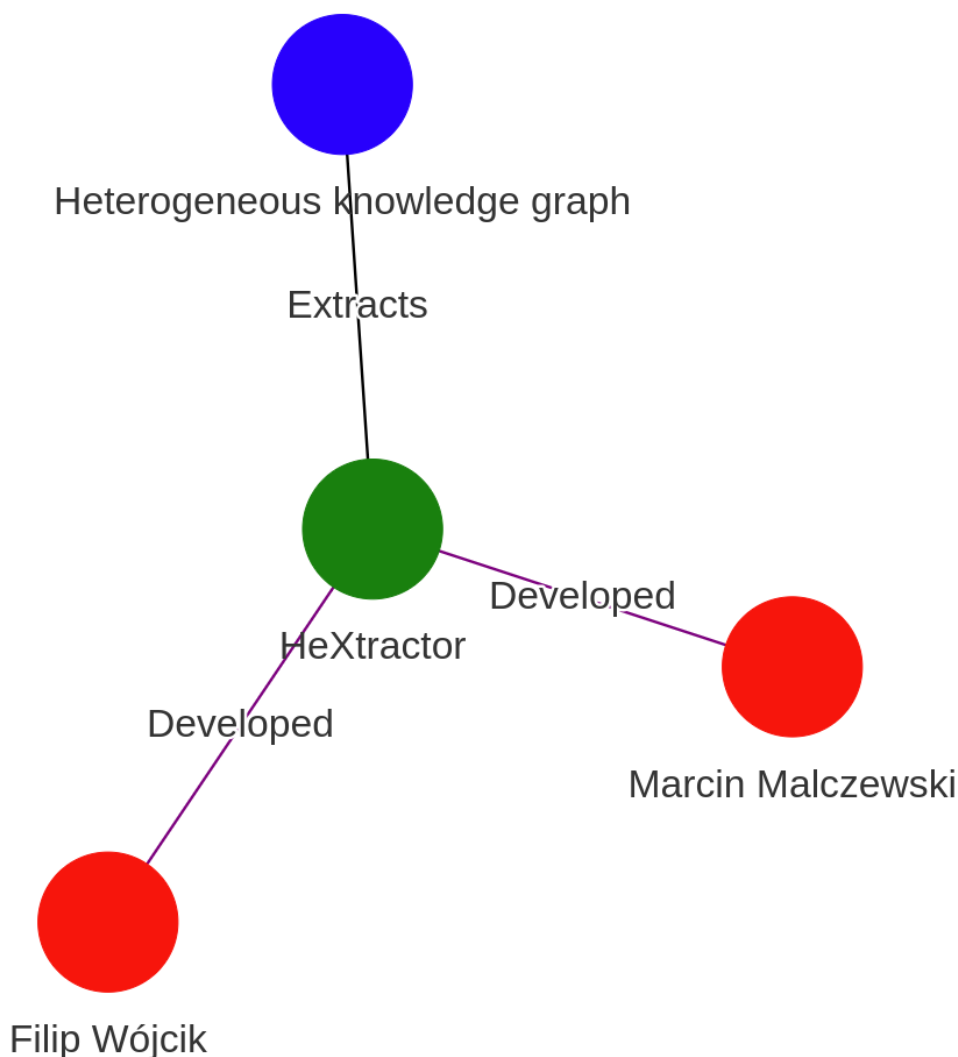


Figure 3: Graph extracted from text.

73 This functionality is particularly valuable for **knowledge graph creation** and **automated document**
74 **analysis**.

75 Visualization

76 HeXtractor leverages **NetworkX** and **PyVis** to provide rich, interactive graph visualizations.
77 Users can configure node types, edge labels, and layout styles, facilitating both interpretability
78 and validation prior to training.

79 Example Use Cases

80 HeXtractor is designed to be domain-agnostic and scalable, accommodating datasets of varying
81 size and complexity. Its capabilities are broadly applicable across numerous research and
82 industrial contexts, including:

- 83 ▪ Banking and fraud detection (Johannessen & Jullum, 2023; Wójcik, 2024)
- 84
- 85 ▪ Recommendation systems (Deng, 2022; Wu et al., 2022)
- 86
- 87 ▪ Biomedical knowledge graphs (Jumper et al., 2021; MacLean, 2021)

88 In each of these areas, HeXtractor enables the integration of multiple data sources—both
 89 structured and unstructured—into cohesive, semantically enriched graph representations. In
 90 the absence of a tool like HeXtractor, this process would typically require significant manual
 91 engineering and carry risks of inconsistency.

92 Documentation

93 Comprehensive documentation, including usage examples and full API reference, is available
 94 at:
 95 <https://hextractor.readthedocs.io/en/latest/>

96 Acknowledgements

97 We gratefully acknowledge the maintainers of **PyTorch Geometric**, **NetworkX**, **LangChain**, and
 98 **pandas** for their foundational contributions. This project received no direct financial support.

99 References

- 100 Chen, X., Liu, Y., Zhang, L., & Kenthapadi, K. (2018). How LinkedIn economic graph bonds
 101 information and product: Applications in LinkedIn salary. *Proceedings of the 24th ACM*
 102 *SIGKDD International Conference on Knowledge Discovery & Data Mining*, 120–129.
- 103 Deng, Y. (2022). Recommender systems based on graph embedding techniques: A review. In
 104 *IEEE Access* (Vol. 10). <https://doi.org/10.1109/ACCESS.2022.3174197>
- 105 Fey, M., & Lenssen, J. E. (2019). *Fast Graph Representation Learning with PyTorch Geometric*.
 106 https://github.com/pyg-team/pytorch_geometric
- 107 Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message
 108 passing for quantum chemistry. *International Conference on Machine Learning*, 1263–1272.
- 109 Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). Heterogeneous graph transformer. *Proceedings*
 110 *of the Web Conference 2020*, 2704–2710.
- 111 Johannessen, F., & Jullum, M. (2023). Finding money launderers using heterogeneous graph
 112 neural networks. *arXiv: 2307.13499*.
- 113 Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool,
 114 K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A.,
 115 Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis,
 116 D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596.
 117 <https://doi.org/10.1038/s41586-021-03819-2>
- 118 MacLean, F. (2021). Knowledge graphs and their applications in drug discovery. *Expert*
 119 *Opinion on Drug Discovery*, 16(9), 1057–1069.
- 120 Mitra, R., Dongre, A., Dangare, P., Goswami, A., & Tiwari, M. K. (2024). Knowledge graph
 121 driven credit risk assessment for micro, small and medium-sized enterprises. *International*
 122 *Journal of Production Research*, 62(12), 4273–4289.
- 123 Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., & Taylor, J. (2019). Industry-scale
 124 knowledge graphs: Lessons and challenges: Five diverse technology companies show how

- 125 it's done. *Queue*, 17(2), 48–75.
- 126 Shi, C. (2022). Heterogeneous graph neural networks. *Graph Neural Networks: Foundations,*
127 *Frontiers, and Applications*, 351–369.
- 128 Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao,
129 T., He, T., Karypis, G., Li, J., & Zhang, Z. (2019). Deep graph library: A graph-centric,
130 highly-performant package for graph neural networks. *arXiv Preprint arXiv:1909.01315*.
- 131 Wang, X., Ji, H., Cui, P., Yu, P., Shi, C., Wang, B., & Ye, Y. (2019). Heterogeneous graph
132 attention network. *The World Wide Web Conference*, 2022–2032. [https://doi.org/10.](https://doi.org/10.1145/3308558.3313562)
133 [1145/3308558.3313562](https://doi.org/10.1145/3308558.3313562)
- 134 Wójcik, F. (2024). An analysis of novel money laundering data using heterogeneous graph
135 isomorphism networks. FinCEN files case study. *Econometrics. Ekonometria. Advances in*
136 *Applied Data Analytics*, 28, 32–49.
- 137 Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph neural networks in recommender
138 systems: A survey. *ACM Computing Surveys*, 55. <https://doi.org/10.1145/3535101>
- 139 Yang, C., Xiao, Y., Zhang, Y., Sun, Y., & Han, J. (2020). Heterogeneous network representation
140 learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge*
141 *and Data Engineering*, 34, 4854–4873. <https://doi.org/10.1109/TKDE.2020.3045924>

DRAFT