




HeXtractor: Extracting Heterogeneous Graphs from Structured and Textual Data for Graph Neural Networks

Filip Wójcik ¹ and Marcin Malczewski²

1 Wrocław University of Economics and Business, Wrocław, Poland 2 Diveapps, Wrocław, Poland

DOI: [10.21105/joss.08057](https://doi.org/10.21105/joss.08057)

Software

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

Editor: [Nikoleta Glynatsi](#) 

Reviewers:

- [@jboynyc](#)
- [@cjbarrie](#)

Submitted: 26 March 2025

Published: 19 June 2025

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 both structured tabular data and unstructured textual content into heterogeneous graph representations suitable for Graph Neural Networks (GNNs). Fully compatible with the PyTorch Geometric (PyG) framework ([Fey & Lenssen, 2019](#)), HeXtractor offers a streamlined, high-level interface for defining entities (nodes), relationships (edges), and associated metadata across diverse data modalities.

Graph Neural Networks have gained significant traction due to the rise of the Message Passing Neural Network (MPNN) paradigm ([Gilmer et al., 2017](#)). Within this context, heterogeneous graphs, which accommodate multiple node and edge types, have emerged as powerful tools in fields such as recommendation systems, fraud detection, and knowledge representation ([Shi, 2022](#); [Yang et al., 2020](#)). Modern architectures—including Heterogeneous Graph Transformers ([Hu et al., 2020](#)) and Heterogeneous Graph Attention Networks ([X. Wang et al., 2019](#))—are specifically optimized to exploit the semantic richness of these graph structures.

One prominent application of heterogeneous graphs lies in knowledge graph construction, which models intricate real-world relationships. These structures have found use across various industries, including employment matching ([Chen et al., 2018](#); [Noy et al., 2019](#)) and credit risk evaluation ([Mitra et al., 2024](#)).

Despite their versatility, constructing heterogeneous graphs is often time-consuming and error-prone. HeXtractor addresses this challenge by providing a standardized, automated framework to convert structured and unstructured data into formats compatible with GNNs, with optional support for large language models (LLMs) to extract graph structures from text.

Statement of Need

A heterogeneous graph can be formally represented as a tuple $G = (V, E)$, where V and E denote 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 are sets of node and edge types ([Shi, 2022](#)). These structures enable modeling of both structural and semantic diversity inherent in complex datasets.

Although libraries such as PyG ([Fey & Lenssen, 2019](#)) and DGL ([M. Wang et al., 2019](#)) offer powerful learning tools for heterogeneous graphs, they lack comprehensive support for the graph construction process—especially when dealing with diverse, multi-source datasets. As a result, researchers often resort to custom scripts, introducing inconsistencies and reducing reproducibility.

HeXtractor fills this gap by offering:

- A declarative interface for defining node and edge schemas;
- LLM integration for extracting graph structures from natural language via LangChain-compatible GraphDocument objects;
- Schema validation and consistency checking;
- Interactive graph visualization;
- Seamless export to PyG's HeteroData format.

Originally developed as part of the HexGIN project (Wójcik, 2024), which focused on financial transaction analysis, HeXtractor has since evolved into a domain-agnostic framework for heterogeneous graph extraction.

Features and Usage

HeXtractor supports the construction of heterogeneous graphs from both structured tabular datasets and unstructured textual sources. It includes built-in visualization capabilities (via PyVis) and is fully interoperable with the PyTorch Geometric framework.

Structured Data Extraction

HeXtractor supports both single-table and multi-table data processing. In a single-table mode, each row encodes relationships among entities defined by columns. Users define:

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

This results in a PyG-compatible HeteroData object, ready for downstream analysis.

Company ID	No. employees	Company revenue	Employee ID	Employee position	Employee Age
1	100	1000	0	0	25
1	100	1000	1	1	35
...
2	5000	100000	6	4	31

Given the exemplary table above, HeXtractor outputs the following heterogeneous graph:

```
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] }  
)
```

Graph visualization is interactive, with support for customized labels and color schemes to enhance interpretability.

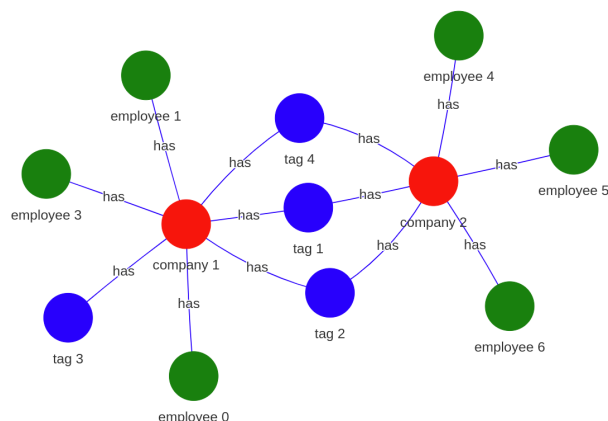


Figure 1: Graph extracted from structured data.

In **multi-table mode**, users define GraphSpecs to merge entity and relationship tables into a unified graph. This approach is well suited for complex data structures with multiple interconnected entities.

For instance, if the dataset from previous table is split into separate company, employee, and tags tables, along with join tables for relationships (as shown in Figure 2), HeXtractor will construct an equivalent HeteroData object—regardless of the original data layout.

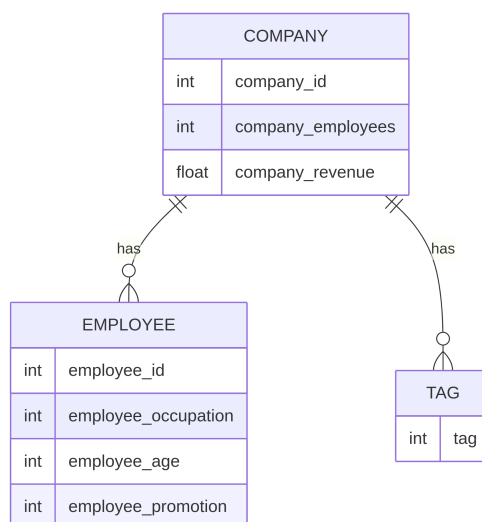


Figure 2: Entity relationship diagram.

Text-Based Graph Extraction

HeXtractor also enables semantic graph construction from natural language, using LLMs via **LangChain**. The process involves:

1. Feeding input text to an LLM;
2. Receiving a GraphDocument containing nodes and relationships;
3. Converting it into a HeteroData object.

For example, the input:

Marcin Malczewski and Filip Wójcik are data scientists who developed HeXtractor. It helps in extraction of heterogeneous knowledge graphs from various data sources.

is transformed into the following HeteroData object:

```
HeteroData(  
  Person={ x=[2, 1] },  
  Library={ x=[1, 1] },  
  Graph={ x=[1, 1] },  
  (Library, Extracts, Graph)={ edge_index=[2, 1] },  
  (Person, Developed, Library)={ edge_index=[2, 2] }  
)
```

Next, it can be visualized as in Figure 3.

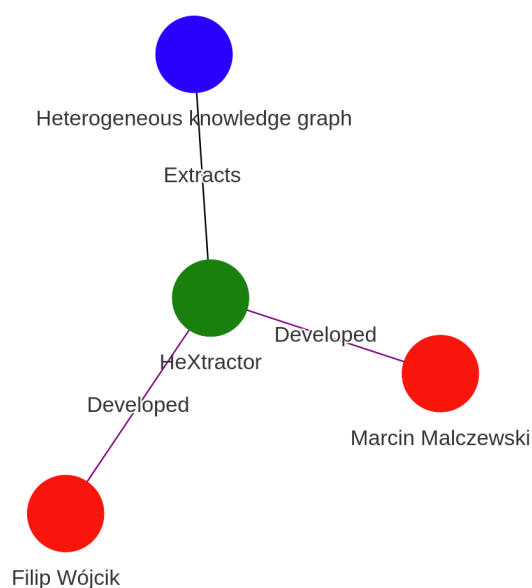


Figure 3: Graph extracted from text.

This feature can be especially useful for automated document analysis and knowledge graph generation.

Example Use Cases

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

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

In these contexts, HeXtractor facilitates the integration of structured and unstructured data into coherent, semantically enriched graph representations. Without such tooling, this process

would typically require extensive manual engineering and be prone to inconsistencies.

Documentation

Comprehensive documentation, including usage examples and full API reference, is available at: [the official website](#).

Acknowledgements

This project did not receive any direct financial support.

References

- Chen, X., Liu, Y., Zhang, L., & Kenthapadi, K. (2018). How LinkedIn economic graph bonds information and product: Applications in LinkedIn salary. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 120–129. <https://doi.org/10.1145/3219819.3219921>
- Deng, Y. (2022). Recommender systems based on graph embedding techniques: A review. In *IEEE Access* (Vol. 10). <https://doi.org/10.1109/ACCESS.2022.3174197>
- Fey, M., & Lenssen, J. E. (2019). *Fast Graph Representation Learning with PyTorch Geometric*. <https://doi.org/10.48550/arXiv.1903.02428>
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. *International Conference on Machine Learning*, 1263–1272. <https://doi.org/10.48550/arXiv.1704.01212>
- Hu, Z., Dong, Y., Wang, K., & Sun, Y. (2020). Heterogeneous graph transformer. *Proceedings of the Web Conference 2020*, 2704–2710. <https://doi.org/10.1145/3366423.3380027>
- Johannessen, F., & Jullum, M. (2023). Finding money launderers using heterogeneous graph neural networks. *arXiv: 2307.13499*. <https://doi.org/10.48550/arXiv.2307.13499>
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596. <https://doi.org/10.1038/s41586-021-03819-2>
- MacLean, F. (2021). Knowledge graphs and their applications in drug discovery. *Expert Opinion on Drug Discovery*, 16(9), 1057–1069. <https://doi.org/10.1080/17460441.2021.1910673>
- Mitra, R., Dongre, A., Dangare, P., Goswami, A., & Tiwari, M. K. (2024). Knowledge graph driven credit risk assessment for micro, small and medium-sized enterprises. *International Journal of Production Research*, 62(12), 4273–4289. <https://doi.org/10.1080/00207543.2023.2257807>
- Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., & Taylor, J. (2019). Industry-scale knowledge graphs: Lessons and challenges: Five diverse technology companies show how it's done. *Queue*, 17(2), 48–75. <https://doi.org/10.1145/3329781.3332266>
- Shi, C. (2022). Heterogeneous graph neural networks. *Graph Neural Networks: Foundations, Frontiers, and Applications*, 351–369. https://doi.org/10.1007/978-981-16-6054-2_16
- Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., & Zhang, Z. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv Preprint arXiv:1909.01315*.

<https://doi.org/10.48550/arXiv.1909.01315>

Wang, X., Ji, H., Cui, P., Yu, P., Shi, C., Wang, B., & Ye, Y. (2019). Heterogeneous graph attention network. *The World Wide Web Conference*, 2022–2032. <https://doi.org/10.1145/3308558.3313562>

Wójcik, F. (2024). An analysis of novel money laundering data using heterogeneous graph isomorphism networks. FinCEN files case study. *Econometrics. Ekonometria. Advances in Applied Data Analytics*, 28, 32–49. <https://doi.org/10.15611/eada.2024.2.03>

Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph neural networks in recommender systems: A survey. *ACM Computing Surveys*, 55. <https://doi.org/10.1145/3535101>

Yang, C., Xiao, Y., Zhang, Y., Sun, Y., & Han, J. (2020). Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 34, 4854–4873. <https://doi.org/10.1109/TKDE.2020.3045924>