World Journal of Cyber Data Science Research (WJCDSR)
ISSN (Online): 3144-508X
Volume-1 Issue-2 || Apr – Jun 2024 || PP. 1-5
https://wjcdsr.org

Using Neo4j for Knowledge Graph Construction in Enterprise NLP

DOI: https://doi.org/10.63345/wjcdsr.v1.i2.201

Justin Rajakumar Maria Thason

BCBSA - Ajace, Inc

AI Research, Innovation & Technical Publications

http://www.wjcdsr.org/ || Vol. No. 2 (2024): June Issue

Date of Submission: 03-05-2024	**Date of Acceptance:** 17-05-2024	**Date of Publication:** 07-06-2024

Abstract : Unstructured text contains significant knowledge—contracts, support tickets, policy documents—that makes cross-document factual queries difficult. This research describes a pipeline that constructs an enterprise knowledge graph in Neo4j from 12,400 documents using transformer-based named entity recognition, relation extraction, and entity resolution. Extraction, performance, and question-answering capabilities of the pipeline were examined. Fine-tuned recognizer achieved a micro F1 score of 0.91, relation extraction a macro/average F1 of 0.83, and entity resolution a pairwise F1 of 0.92, aggregating 241,000 mentions into 184,200 nodes interrelated by 612,500 relationships. Queries of five hops in Neo4j were executed in 171 ms, compared to 118 seconds with relational joins. Hybrid graph-and-vector retrieval for 200 multi-hop questions had a correct response rate of 79.5%, compared to 50.5% for vector retrieval ($p < 0.001$). Thus, Neo4j-based graph construction turns unstructured enterprise text into a queryable resource that facilitates multi-hop reasoning.

Keywords: knowledge graph; Neo4j; Cypher; enterprise NLP; named entity recognition; relation extraction; entity resolution; GraphRAG

Introduction

Often, business-related questions are multi-document—they may ask which suppliers are bound by clauses that contradict a new regulation, which products were involved in incidents managed by a certain team, or which customers have contacts with a terminated partner. Answering such questions requires linking mentions of the same organizations, persons, and products across thousands of documents and following chains of relations.

Vector-based retrieval retrieves text passages that are similar to a query without representing relations between entities, hence not allowing multi-step inference. Knowledge graph encodes entities as nodes and relations as edges, enabling multi-step traversal. Such knowledge graphs are often stored in graph databases, including Neo4j, accessed using Cypher language.

This research will introduce a construction pipeline, measure its performance at each stage, compare query performance to the relational approach, and evaluate usefulness for question answering. It aims to give practitioners the empirical evidence about the quality, speed, and utility of an enterprise knowledge graph constructed in the proposed way.

@2024. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on www.wjcdsr.org

World Journal of Cyber Data Science Research (WJCDSR)
ISSN (Online): 3144-508X
Volume-1 Issue-2 || Apr – Jun 2024 || PP. 1-5
https://wjcdsr.org

Literature Review

Surveys on knowledge graphs were done by Hogan et al. (2021), analyzing the models, construction, quality assessment, and use cases of knowledge graphs, and by Ji et al. (2022), covering representation, acquisition, and application of knowledge graphs. Nickel et al. (2016) reviewed approaches to relational machine learning applied to knowledge graphs, including link prediction. Robinson et al. (2015) introduce graph databases and property-graph model, while Cypher language is formalized by Francis et al. (2018).

Knowledge graph construction presupposes information extraction. Models pretrained on large datasets of text and represented by transformers set new benchmarks in sequence labeling and classification after fine-tuning (Devlin et al., 2019) and are used for named entity recognition and relation extraction. The task of recognizing whether certain records refer to the same real-world entity—entity resolution—is studied since long ago and reviewed by Christophides et al. (2020), highlighting the role of blocking in avoiding pair-wise comparisons and increasing popularity of learned measures.

Retrieval-augmented generation connects the output of a language model to relevant documents retrieved (Lewis et al., 2020); however, vector retrieval is limited to superficial similarity of passages and does not guarantee the connectivity necessary for multi-hop reasoning. Methods enhancing vector retrieval with graph structure are proposed as means to improve multi-hop reasoning, according to Pan et al. (2024) roadmap of the integration of large language models with knowledge graphs.

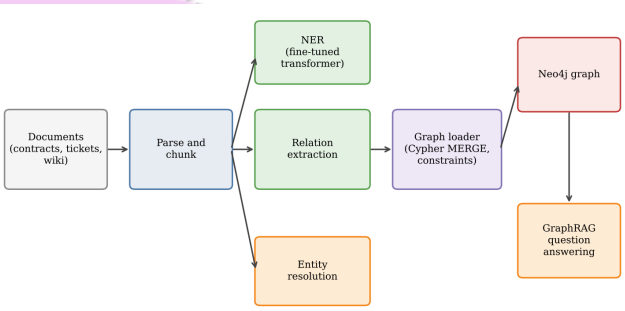
Despite these advances, there is little empirical evidence regarding full pipeline in enterprises, especially concerning query performance of graph databases compared to relational joins. The current study will give such evidence on a realistic corpus.

Research Methodology

Corpus and pipeline

The corpus includes 12,400 documents: 3,100 contracts, 5,200 support tickets, 2,400 policy and wiki pages, and 1,700 internal reports. Figure 1 shows the pipeline. Documents are parsed and segmented into chunks. A transformer fine-tuned on 4,000 annotated chunks identifies five types of entities (organization, person, product, contract clause, and location), with the help of a gazetteer of known products and customers. A secondary model classifies relations between entity pairs in the same or consecutive sentences into eight categories, e.g., SUPPLIES, GOVERNED_BY, and RAISED_BY. Entity resolution uses embedding-based blocking followed by pairwise classifier, aggregating mentions into canonical entities with references to the source chunks. A loader imports the results into Neo4j with uniqueness constraints and MERGE statements, thus ensuring idempotence.

Figure 1. Knowledge graph construction pipeline in Neo4j and answer generation over it.



Evaluation

Extraction quality is evaluated on test sets held-out in annotations: 2,000 entity mentions and 1,200 candidate relations, with the use of baselines explained in results section. Query performance is measured for 100 queries per hop count (from 2 to 5), comparing a Neo4j graph database with equivalent normalized PostgreSQL schema employing recursive joins on the same hardware. Question answering is evaluated for 200 questions comprising 50 examples of each of four reasoning depths (1, 2, 3, and 4 or more hops), formulated by domain specialists with verified answers. Three approaches are evaluated: (i) vector retrieval over partitioned chunks, (ii) graph-only retrieval using generated Cypher queries, and (iii) hybrid retrieval based on graph traversal and providing connected entities along with the sources to language model.

Analysis

@2024. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on www.wjcdsr.org

World Journal of Cyber Data Science Research (WJCDSR)
ISSN (Online): 3144-508X
Volume-1 Issue-2 || Apr – Jun 2024 || PP. 1-5
https://wjcdsr.org

Extraction performance is measured by precision, recall, and F1 metrics. Performance comparison uses Welch's t-test, and performance speed-ups are reported. Accuracy of answers is measured using two-proportions z-test and chi-square test.

Results

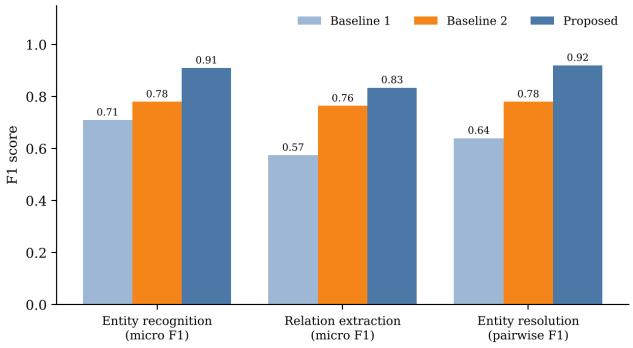
Extraction and entity resolution

Table 1. Named entity recognition performance by entity type

Entity type	Precision	Recall	F1
Organisation	0.94	0.93	0.93
Person	0.96	0.95	0.95
Product	0.89	0.86	0.87
Contract clause	0.87	0.84	0.85
Location	0.93	0.92	0.92

The proposed recognizer achieves a micro F1 of 0.91, outperforming CRF (0.78), off-the-shelf recognizer (0.71), and zero-shot prompting of language model (0.80). It gives the best results for organizations and persons (F1=0.93 and 0.95, respectively) and the worst results for contract clauses (F1=0.85), because of the ill-defined boundaries. Concerning relation extraction, the proposed model has F1=0.83 (precision 0.86, recall 0.81), compared to 0.57 for pattern-based extraction that has high precision (0.91) but fails to find more than half of the relations (recall 0.42) and 0.76 for zero-shot prompting (Figure 2).

Figure 2. F1 scores for entity recognition, relation extraction, and entity resolution (baselines correspond to the first and second approaches mentioned in the text).



Entity resolution achieves a pairwise F1 of 0.92, compared to 0.64 for exact string match and 0.78 for fuzzy match, aggregating 241,000 mentions into 184,200 canonical nodes (24% reduction) and establishing 612,500 relations between them.

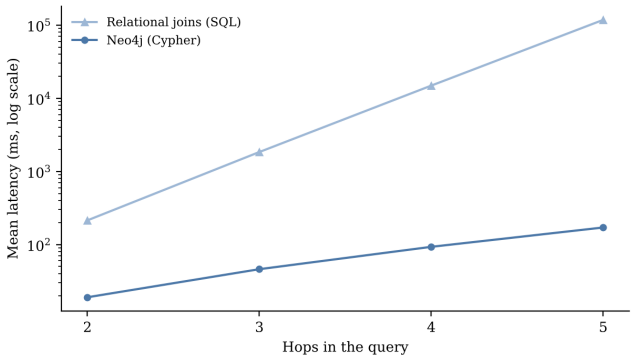
Query performance

Table 2. Mean query latency (ms, mean ± SD, 100 queries per hop count)

Hops	Relational joins	Neo4j (Cypher)	Speed-up	p
2	214 ± 58	19 ± 5	11×	<0.001
3	1,840 ± 420	46 ± 12	40×	<0.001
4	14,900 ± 3,800	93 ± 27	160×	<0.001
5	118,000 ± 31,000	171 ± 49	690×	<0.001

The latency grows exponentially with hops for relational joins, but modestly for Neo4j (Figure 3). At distance of two hops, graph database performs eleven times faster, at five hops—it performs 690 times faster (171 ms vs 118 s; p<0.001). The reason is index-free adjacency, making the cost of traversal depend on the neighborhood visited, not on the table size.

Figure 3. Mean query latency as a function of hop count for relational joins and Neo4j (logarithmic scale).



Question answering

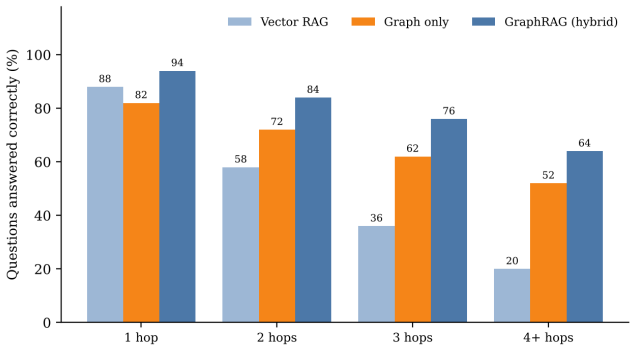
Accuracy of answers differs between the approaches ($\chi^2(2)=37.5$, $p<0.001$). The hybrid approach gives 159 correct answers out of 200 (79.5%), compared to 134 (67.0%) for graph-only retrieval (p=0.005) and 101 (50.5%) for vector retrieval (p<0.001). Advantage of

@2024. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on www.wjcdsr.org

World Journal of Cyber Data Science Research (WJCDSR)
ISSN (Online): 3144-508X
Volume-1 Issue-2 || Apr – Jun 2024 || PP. 1-5
https://wjcdsr.org

hybrid approach grows with depth of reasoning: for one-hop questions the approaches differ only by six correct answers, but for four-or-more hops hybrid approach gives 64% correct answers compared to 20% for vector retrieval (Figure 4).

Figure 4. Accuracy of question answering as a function of reasoning depth for vector retrieval, graph-only retrieval, and hybrid retrieval.



Discussion

The pipeline showed that relatively small annotated corpus is enough to build a working enterprise graph when fine-tuned models are combined with gazetteer and entity resolution. Superiority of fine-tuning vs zero-shot prompting for extraction (especially domain-specific clause types) confirms the efficiency of supervised transformer models in sequence labeling tasks (Devlin et al., 2019). At the same time, high precision but low recall of pattern-based extraction illustrate classical trade-off of rule-based methods.

Entity resolution was critical for our pipeline. Without consolidation, the graph would contain duplicates of the same real-world entity, preventing traversals; hence, the importance of blocking and learned matching in entity resolution literature (Christophides et al., 2020). 24% reduction of nodes indicates that mentions are fragmented within enterprise text.

Findings about the performance and accuracy are complementary. Graph traversal lowers the cost of multi-hop access, as can be expected from native graph storage and Cypher pattern-matching paradigm (Robinson et al., 2015; Francis et al., 2018). Hybrid approach provides connected evidence to the language model, addressing limitations of vector retrieval in multi-step questions (Lewis et al., 2020; Pan et al., 2024). Lagging of graph-only approach was caused by the incompleteness of extracted relations and errors in Cypher; hence, sources still have some value. Assessment of quality and provenance (highlighted by Hogan et al., 2021) are therefore critical.

Limitations

The corpus was produced by a single organization and is in English; the schemas of entities and relations reflect the domains of this organization, which limits the generalization of the findings to other schemas or languages. Test set annotation is performed by a small team with imperfect agreement on clause boundaries. The relational baseline is normalized; better indexing could potentially close the performance gap at lower hops. Questions are produced by the staff familiar with the content of the graph. Security concerns, privacy concerns, and updating the graph with new documents are not addressed.

Conclusion

Pipeline including fine-tuning, entity resolution, and idempotent import yields a Neo4j knowledge graph containing 184,200 entities and 612,500 relations from 12,400 documents, with F1 scores of 0.91, 0.83, and 0.92 for recognition, relations, and resolution, respectively. Multi-hop queries perform up to 690 times faster than relational joins, and hybrid graph-and-vector question-answering raises the accuracy to 79.5% from 50.5%. For enterprises dealing with knowledge in document form, the graph construction should be considered as the cornerstone of search, analysis, and trustworthy generative applications. Further work should focus on incremental updates, access control, and multilingual corpora.

References

1. Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K. (2020). An overview of end-to-end entity resolution for big data. ACM Computing Surveys, 53(6), Article 127.
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT, 4171–4186.
3. Francis, N., Green, A., Guagliardo, P., et al. (2018). Cypher: An evolving query language for property graphs. Proceedings of the ACM SIGMOD International Conference on Management of Data, 1433–1445.

@2024. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on www.wjcdsr.org

World Journal of Cyber Data Science Research (WJCDSR)
ISSN (Online): 3144-508X
Volume-1 Issue-2 || Apr – Jun 2024 || PP. 1-5
https://wjcdsr.org

4. Hogan, A., Blomqvist, E., Cochez, M., et al. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), Article 71.
5. Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems, 33(2), 494–514.
6. Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.
7. Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. Proceedings of the IEEE, 104(1), 11–33.
8. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering, 36(7), 3580–3599.
9. Robinson, I., Webber, J., & Eifrem, E. (2015). Graph databases: New opportunities for connected data (2nd ed.). O'Reilly Media.

WJCDSR
ISSN: 3144-508X (US)

@2024. This is an open access article distributed under the terms of the Creative Commons License [CC BY NC 4.0] and is available on www.wjcdsr.org