

Indexed graph databases for querying rich TEI annotation

Piotr Pezik, University of Łódź

The need to store multiple levels (morphological, syntactic, semantic) and multiple interpretations (generated by multiple tools used at the same level) of linguistic metadata can significantly increase the complexity of corpus annotation. While this complexity has been demonstrated to be manageable by TEI standards, an open question remains of the choice of optimal technologies for the implementation of scalable storage and search engines which allow their users to take full advantage of the richness of such annotation. Both relational databases and customised full-text information retrieval architectures have been used to enable the querying of richly annotated corpora. Typical limitations of these solutions are related to problems with scalability and complexity of representation respectively (cf. Pezik 2012).

Graph databases are a relatively recent, though increasingly popular No-SQL (not-only-SQL) technology which has been found to naturally serve the requirements of storing and searching highly linked data with multiple degrees of separation. This paper demonstrates how complex linguistic annotation originally stored in TEI formats can be transformed into a performant, searchable graph database. Two specific implementations of graph database storage and retrieval systems for querying richly annotated corpus data collections are discussed:

- a) A searchable graph database representing the TEI P5 manually annotated 1-million word subcorpus of the National Corpus of Polish (NKJP)
- b) A distributed graph database backend for the multi-foundry annotation generated in the KorAP project. Although non-TEI, KorAP annotation may be taken as the next step in complexity, to which the development of stand-off TEI annotation may aspire (cf. Bański 2010).

In the case of the NKJP data, a need arises to enable the querying of multiple levels of textual and linguistic annotation stored in a TEI-compliant format, including segmental, morphological, shallow syntactic, named-entity and word-sense annotation with prospective extensions to accommodate deep syntactic annotation. In KorAP, segmental, morphological, syntactic (including chunked, dependency and constituency) annotation is generated by different tools simultaneously, thus overlapping to various extents across the underlying primary textual data. The paper shows how these challenging data sets can be modelled and queried using the Neo4J graph database implementation with customized Apache Lucene indexes.

Complex corpus annotation is mapped onto the Neo4J database as an arbitrary multigraph with properties in which vertices represent entities such as words, constituents, chunks, sentences, paragraphs and texts, while edges represent relations between entities, such as dependencies, domination, sequence, subsumption and coincidence as in the case of overlapping span annotations across different foundries in the KorAP data. Both vertices (nodes)

and edges (relations) can have multiple types and properties. Types of nodes and relations can be used to define a loose schema in what is an otherwise largely schemaless database to be used as entry points in graph queries. As an example, all nodes of type WORD_SEGMENT can be used as starting points in a positional concordance query. Properties on nodes and relations can be used to store single values and collections of values for which logical conditions can be specified in subsequent queries. Queries against a corpus graph database can be described as definitions of subgraphs for which starting points in the form of identifiers of nodes and edges are provided.

Two possible implementations of a search engine backend for corpus graph databases are proposed depending on the performance requirements and data size. On the one hand graph databases can be queried directly either in an embedded mode or through REST interfaces. The Cypher query language can be used to query multi-level TEI annotation schemas in question with the Gremlin query language and the core Neo4J graph traversal API as possible alternatives. Large text data collections, on the other hand, need to be indexed first to narrow down the number of potentially relevant graph nodes entered in a search. The latter approach is more problematic due to the need of translating linguistic information formulated in user queries into shallow search queries, which are subsequently converted to 'deep' graph database queries. Both of these approaches rely on the assumption that TEI corpora can be divided into separated graphs. In other words, no hard-coding of edges across different texts is deemed necessary (which does not preclude the possibility of dynamically aggregating text metadata at query time). An arbitrary number of texts can thus be indexed as a separate shard in a distributed graph database.

The current graph database implementations of the NKJP TEI data shows that on average 2.45 nodes, 8.2 properties and 3.9 relationships are necessary to represent one word segment. This estimation is even higher for the KorAP data, due to its multi-foundry structure with independent segmentation and heterogeneous annotation generated by the different tagging and parsing tools used. Therefore a scalable, distributed architecture of the search back-end is recommended for querying rich TEI annotation, in which both shallow and deep queries are dispatched to and collected from a cluster of sharded indexes and separated graph databases in a map-reduce fashion.

References

1. Bański, Piotr, and Adam Przepiórkowski. "Stand-off TEI annotation: the case of the National Corpus of Polish." *Proceedings of the third linguistic annotation workshop*. Association for Computational Linguistics, 2009.
2. Bański, Piotr, et al. "The new IDS corpus analysis platform: Challenges and prospects." *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*. 2012.
3. Bański, Piotr. "Why TEI stand-off annotation doesn't quite work: and why you might want to use it nevertheless." *Proceedings of Balisage: The Markup Conference*. Vol. 5. 2010.
4. Łukasz Degórski and Adam Przepiórkowski. Ręcznie znakowany milionowy podkorpus NKJP. (Manually annotated 1 million-word subcorpus of NKJP). In Adam Przepiórkowski, Mirosław Bańko,

Rafał L. Górski,

5. and Barbara Lewandowska-Tomaszczyk, editors, Narodowy Korpus Języka Polskiego, pages 51–58. Wydawnictwo Naukowe PWN, Warsaw, 2012.
6. Pęzik, P. (2012). Wyszukiwarka PELCRA dla danych NKJP (PELCRA search engine for NKJP). In Narodowy Korpus Języka Polskiego. Pp. 253-279. Wydawnictwo Naukowe PWN, Warsaw
7. Uzar, R. Pęzik, P., Levin E. (2004) Developing relational databases for corpus linguistics In Practical Applications in Language and Computers PALC 2003. Frankfurt am Main, Berlin, Bern, Bruxelles, New York, Oxford, Wien. ISBN 978-3-631-52461-9
8. <http://www.neo4j.org/>
9. <http://lucene.apache.org/>