The Problem

- NLP is being used in more and more areas to perform (automatic) text analysis, e.g., medicine, biology, (digital) humanities, economy, legal theory, etc.
- Data for NLP tasks is often huge and diverse: graph-based models, document-based models (e.g., TEI), or combinations.
- We developed TextImager, a web service based NLP framework for big data (Hemati, Uslu, and Mehler 2016).
- NLP data is stored as XML-files (in our case serialized UIMA Common Analysis Structure UIMA-CAS) (Abrami and Mehler 2018).
- Further processing needs to always deserialize the thousands of XML-files, e.g., running quantitative analysis like TF-IDF and Type-Token-Ratio.
- It is time consuming to add additional annotation layers, or query specific parts.
- We propose a database system to be able to query annotations of documents according to the varying data models.

The question: which existing database based on which paradigm performs best for which NLP tasks?

- NLP data is stored as UIMA-CAS objects, serialized and compressed to XMI files (Grose, Doney, and Brodsky 2002).

Databases

- MySQL: Open-source relational database management system, we use the NLP-related scheme from (Fette, Toepfer, and Puppe 2013).
- MongoDB: Scalable document-oriented NoSQL database, UIMA-CAS objects are serialized into binary-encoded JSON objects.
- BaseX: Light-weight XML document-oriented database, UIMA-CAS objects are serialized into XML documents.
- Cassandra: Column-oriented NoSQL database, every annotation layer of the typesystem uses a separate table.
- Neo4j: Highly performant NoSQL graph database, each input text is represented as a node linking to all its token nodes whose syntagmatic order is mapped by token links.

Storage Performance

![Storage Performance Graph]

Reading Performance

![Reading Performance Graph]

Query Performance

<table>
<thead>
<tr>
<th>Database</th>
<th>POS</th>
<th>Lemma</th>
<th>Morph</th>
<th>Dep</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>1,290.2</td>
<td>24.9</td>
<td>13.8</td>
<td>127.6</td>
</tr>
<tr>
<td>Neo4j</td>
<td>1,287.0</td>
<td>27.5</td>
<td>57.0</td>
<td>185.9</td>
</tr>
<tr>
<td>BaseX</td>
<td>1,299.1</td>
<td>26.3</td>
<td>48.0</td>
<td>132.3</td>
</tr>
<tr>
<td>MySQL</td>
<td>1,490.2</td>
<td>371.4</td>
<td>59.4</td>
<td>2,349.7</td>
</tr>
</tbody>
</table>

Evaluation

- XML: Extremely fast storing (just save the file), but also extremely slow querying.
- Neo4j: Fast storing, reading and relation querying.
- MySQL: Fast attribute querying and reading, but really slow storing.
- MongoDB: Consistently good performance, for all tasks.
- Cassandra: Performs ok, but not usable for our type of queries due to not supporting table joins for relations.
- BaseX: Relatively slow compared to the others.
- This shows no clear winner and depending on the task users should utilize different databases.
- Combination is needed: we introduce a web-based multi-database management system (MDBMS) integrated in TextImager.

MDBMS

- Provides a single interface for querying different databases.
- All data processed by TextImager is stored in multiple databases.
- Projects can simultaneously manage and query a variety of data using these databases:
  - Using the MDBMS interface using our combined databases, or
  - Directly accessing a single database, to use more specialized queries.

References:


www.texttechnologylab.org