

# Text2Scene

# Object Dictionaries

Patrick Schrottenbacher

# Agenda

1. Aufgabenstellung
2. Neo4j
3. NLTK/Wordnet
4. Vorgehensweise
5. Resultat
6. Quellen

# Aufgabenstellung

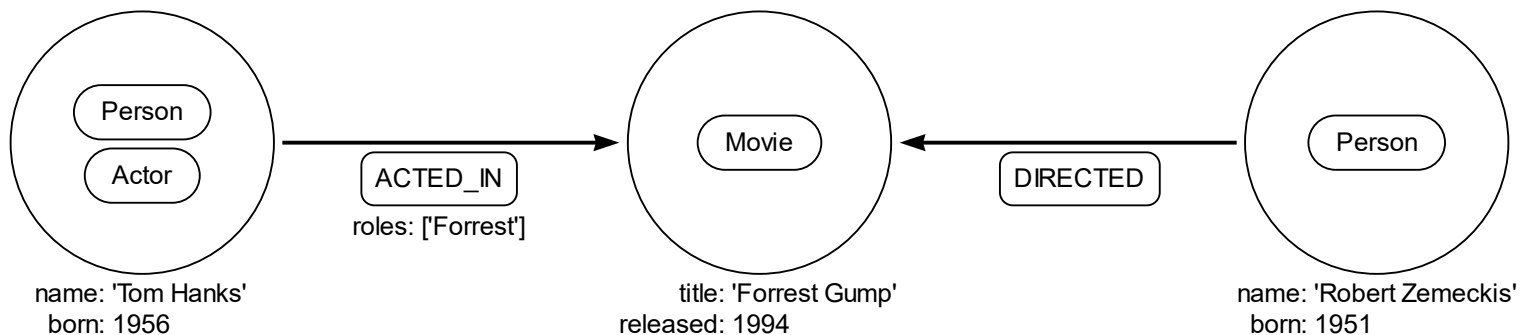
Struktur des WordNet Dictionarys in Neo4j umsetzen.  
Zum extrahieren der Wörter existiert ein python  
package namens NLTK

# Agenda

1. Aufgabenstellung
2. Neo4j
3. NLTK/Wordnet
4. Vorgehensweise
5. Resultat
6. Quellen

# Neo4j / Graphdatenbanken

- Knoten und Kanten
- Spezifisch: Knoten sowie Kanten haben beliebig viele Attribute, sowie labels
- Pros
  - Schnelles querying für (stark) verbundene Datensätze
  - Gut fürs modellieren komplexer/irregulärer Datensätze
- Kons
  - Größerer Speicherplatz verbrauch (im Vergleich zu RDBs)
  - Datenstruktur nicht statisch



# Agenda

1. Aufgabenstellung
2. Neo4j
3. NLTK/Wordnet
4. Vorgehensweise
5. Resultat
6. Quellen

# NLTK/Wordnet

- NLTK: interface für lexikalische Ressourcen
- Wordnet: Lexikarische Datenbank
- Fokus liegt auf Synonymen, Antonymen, Hyponymen etc.

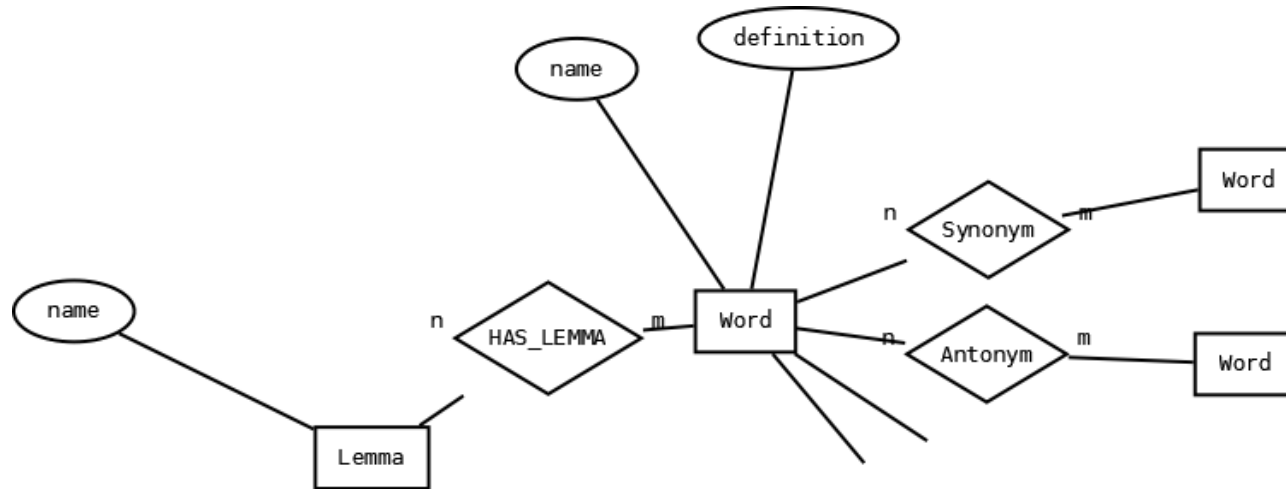
# Agenda

1. Aufgabenstellung
2. Neo4j
3. NLTK/Wordnet
4. Vorgehensweise
5. Resultat
6. Quellen



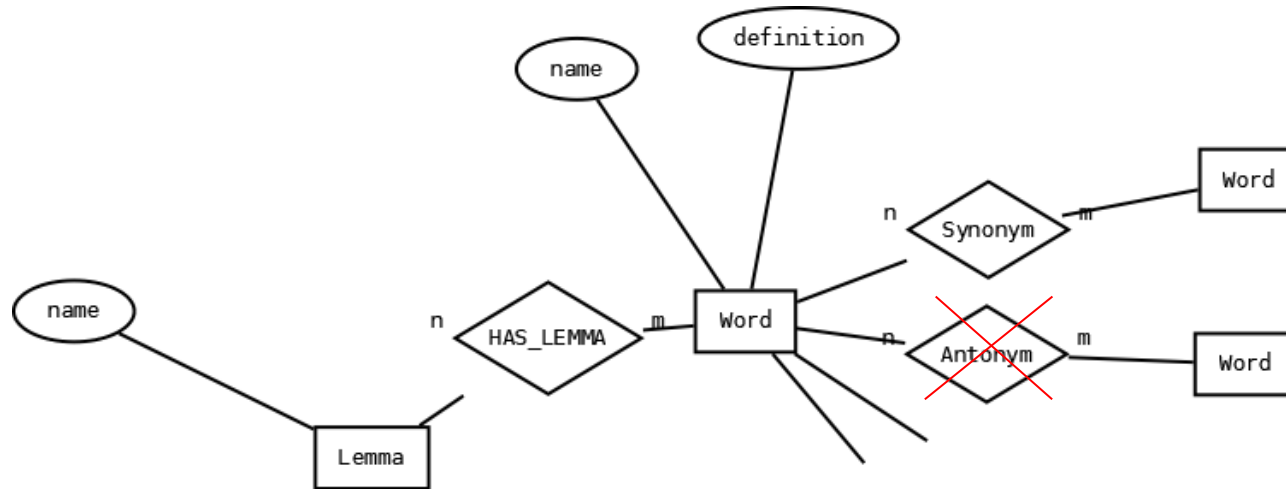
# Vorgehensweise

- Packages: nltk, neomodel (neo4j)



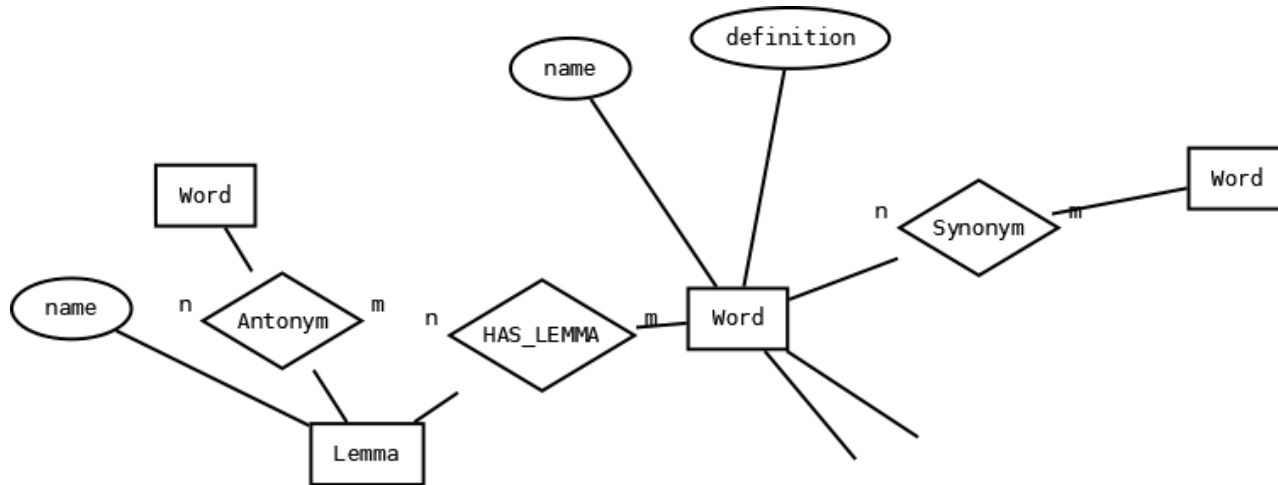
# Vorgehensweise

- Packages: nltk, neomodel (neo4j)



# Vorgehensweise

- Packages: nltk, neomodel (neo4j)



# Vorgehensweise

## Idee 1

Alle Synsets (Wörter) durchgehen und ein-speichern, rekursiv die Beziehungen erstellen falls nötig die Nodes erstellen.

Probleme:

- Langsam, da keine I/O-Optimierung
- Es kann zu sehr tiefen Rekursionen kommen

# Vorgehensweise

## Idee 2

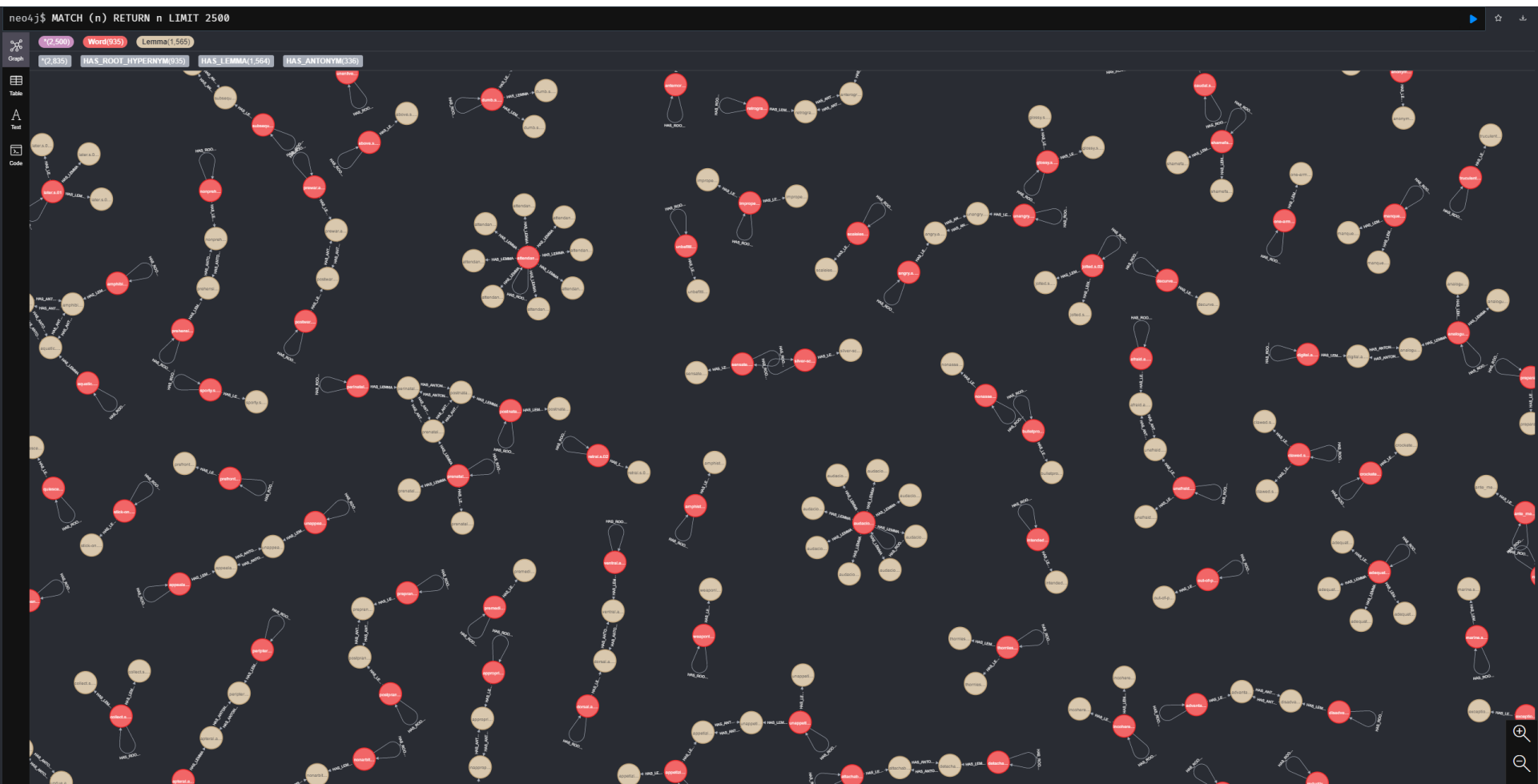
Alle Synsets (Wörter) und deren Lemmas ein-speichern, später die Beziehungen definieren.

- Erlaubt batching zur Reduktion von I/O
- Keine tiefen Rekursionen
- Erlaubt es einfacher updates durchzuführen.

# Agenda

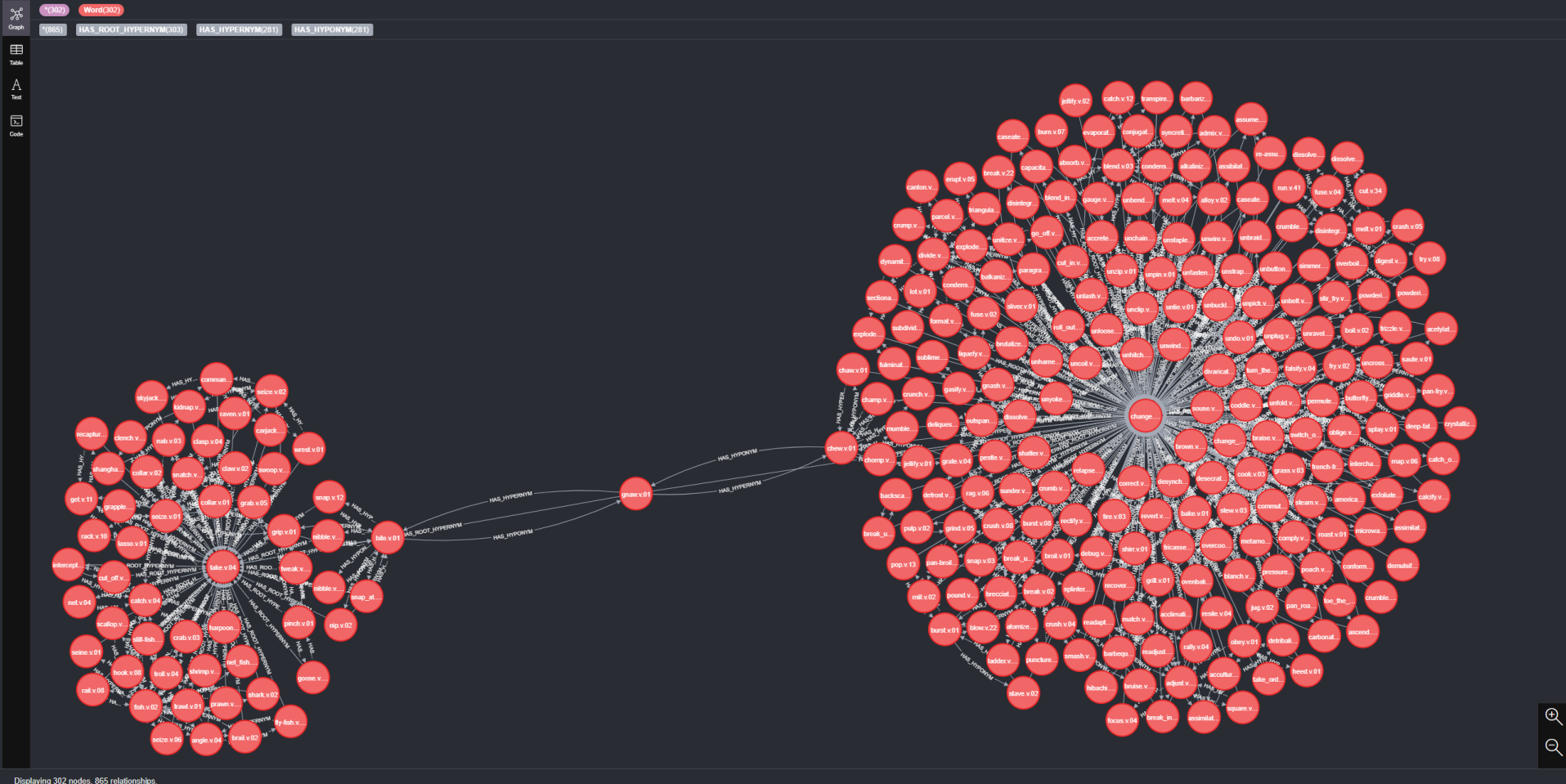
1. Aufgabenstellung
2. Neo4j
3. NLTK/Wordnet
4. Vorgehensweise
5. Resultat
6. Quellen

# Resultat



# Resultat

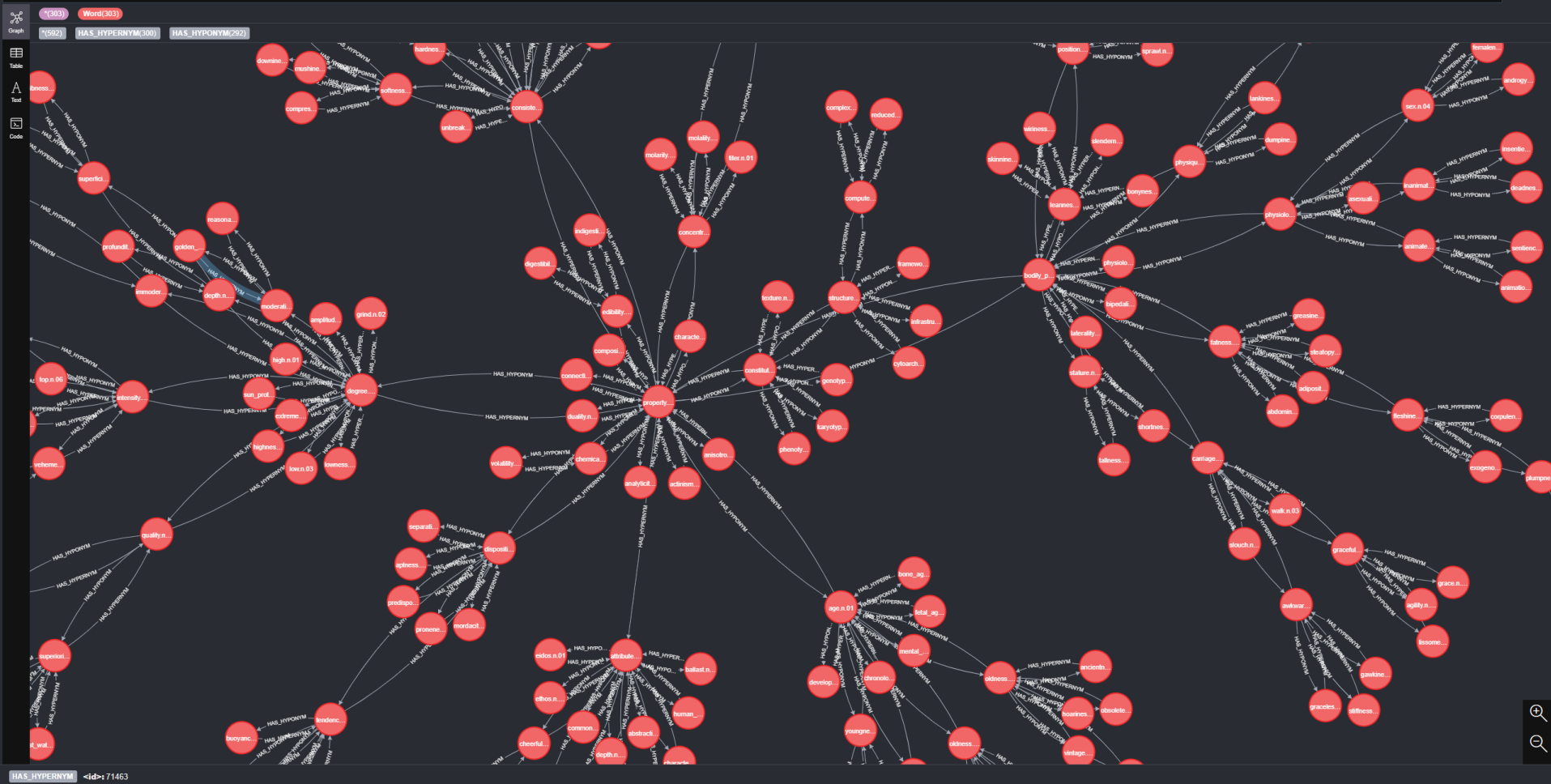
```
neo4j$ MATCH p=(a)-[r:HAS_ROOT_HYPERNYM]->(c) where a <> c and a.name <> "entity.n.01" and c.name <> "entity.n.01" RETURN p LIMIT 300
```





# Resultat

```
neo4j$ MATCH p=(a)-[r:HAS_HYPERNYM]→(c) where a < c and a.name < "entity.n.01" and c.name < "entity.n.01" RETURN p LIMIT 300
```



# Agenda

1. Aufgabenstellung
2. Neo4j
3. NLTK/Wordnet
4. Vorgehensweise
5. Resultat
6. Quellen

# Quellen

- neomodel: [https://neomodel.readthedocs.io/\\_/downloads/en/stable/pdf/](https://neomodel.readthedocs.io/_/downloads/en/stable/pdf/)
- neo4j: <https://neo4j.com/>
- NLTK/wordnet: <https://www.nltk.org/howto/wordnet.html>

Vielen Dank für Ihre  
Aufmerksamkeit!