

# Text2Scene

# Object Dictionaries

Patrick Schrottenbacher

# Agenda

## 1. Aufgabenstellung

## 2. Verlauf

1. Wordnet
2. Wordvector
3. BERT
4. Evaluation

## 3. Final thoughts

## 4. Quellen

# Aufgabenstellung

Erstellung eines Object-dictionaries mithilfe von  
Wordnet, BERT und Wordvectors

# Agenda

1. Aufgabenstellung

2. Verlauf

1. Wordnet

2. Wordvector

3. BERT

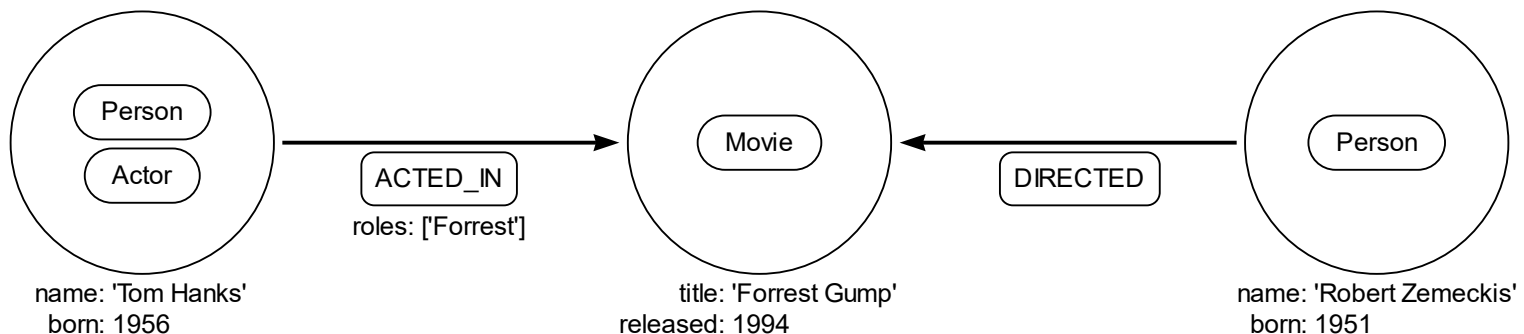
4. Evaluation

3. Final thoughts

4. 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. Verlauf

1. Wordnet

2. Wordvector

3. BERT

4. Evaluation

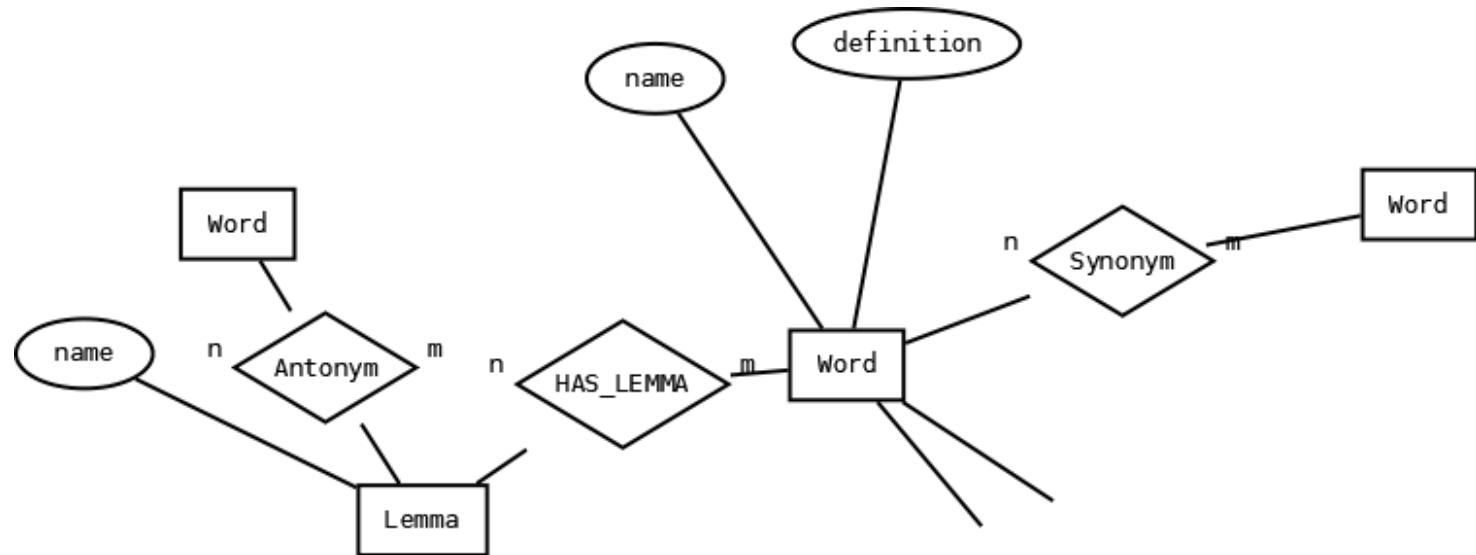
3. Final thoughts

4. Quellen

# NLTK/Wordnet

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

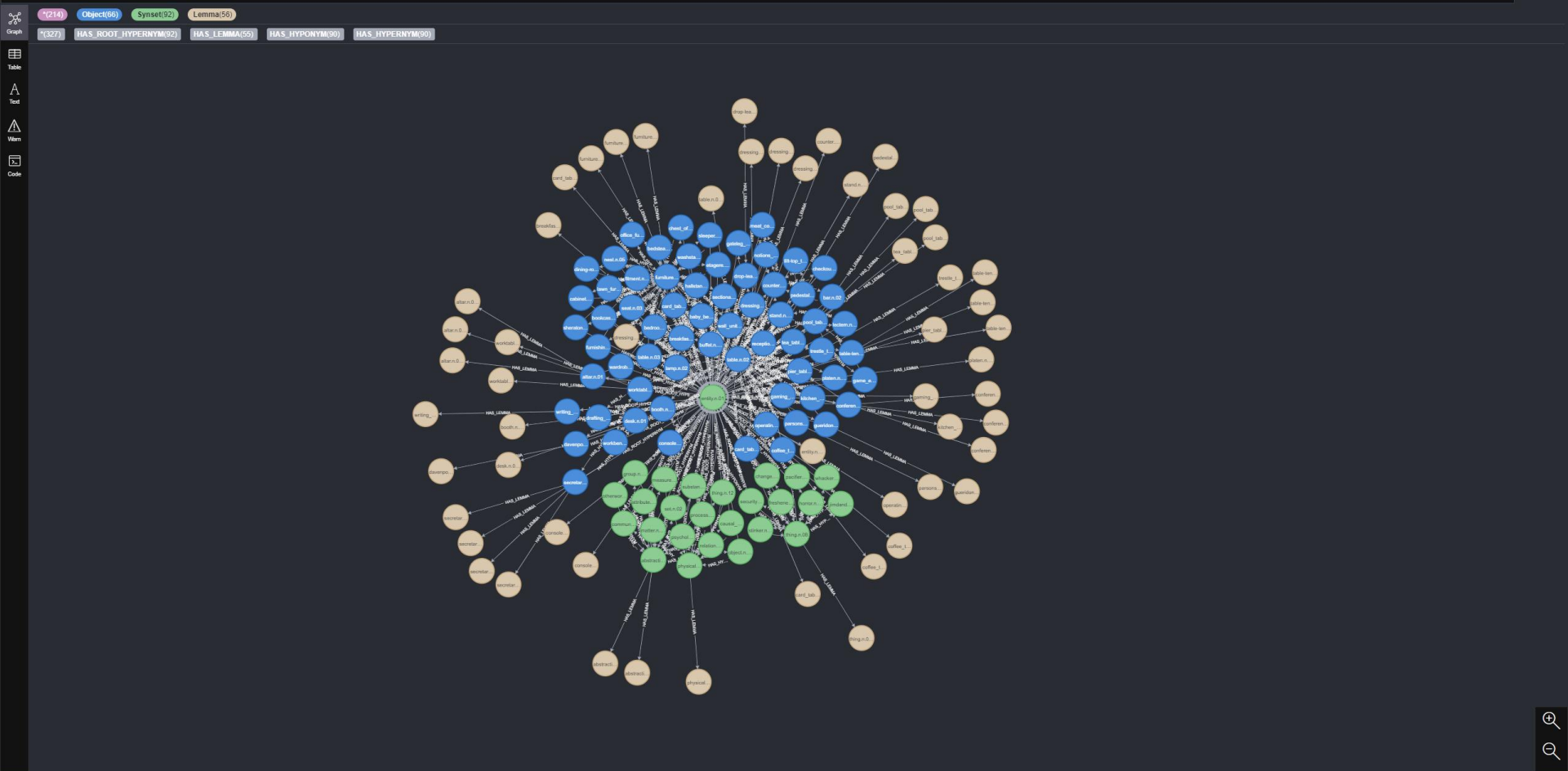
# Datenstruktur





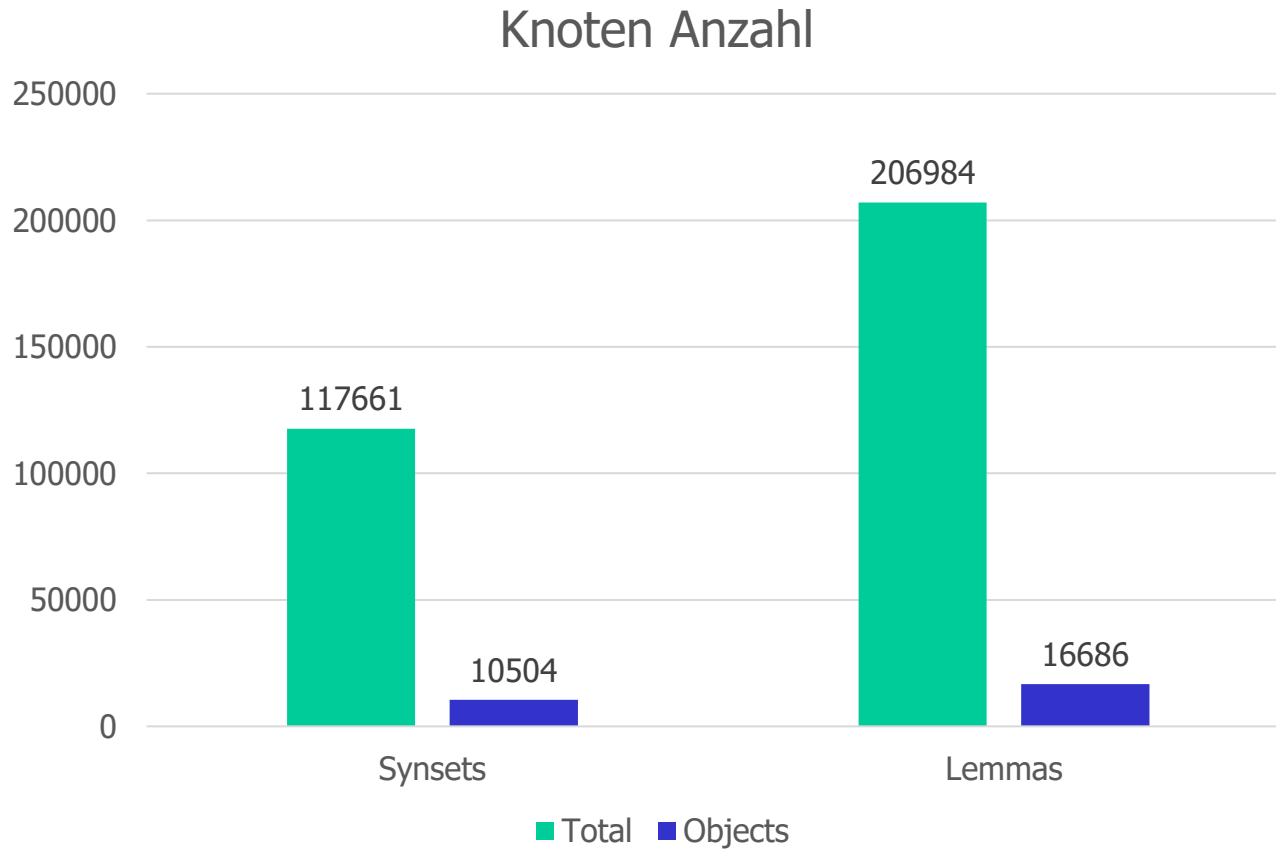
# Objekt Klassifizierung

```
neo4j$ match p = (n:Synset)-[r*1..3]->() where n.name = "desk.n.01" return p limit 1000
```



Displaying 147 nodes, 327 relationships.

# Resultat



# Agenda

1. Aufgabenstellung
2. Verlauf
  1. Wordnet
  2. Wordvector
  3. BERT
  4. Evaluation
3. Final thoughts
4. Quellen

# Wordvector

- FastText: Jedes Wort hat eine Vektorrepräsentation
- Wir können die 10 nächsten Vektorrepräsentationen nehmen und die relation speichern.
- Problem: FastText arbeitet mit „normalen Wörtern“ d.h. es gibt z.B. keine Unterschied zwischen der Farbe „blau“ und „blau sein“
- => Restrukturierung der Datenstruktur

# Restrukturierung

- Jedes Lemma bekommt ein "RootWord" zugeschrieben
- Mehrere Lemma können somit ein gemeinsames "RootWord" besitzen
- Unterteilung dieser in "RootWordObject"
- Wir können nun Wordvector Relationen für unsere "RootWords" einspeichern

# Agenda

1. Aufgabenstellung
2. Verlauf
  1. Wordnet
  2. Wordvector
  3. BERT
  4. Evaluation
3. Final thoughts
4. Quellen

# BERT

- Erstellung von Sätzen welche Informationen über Objekte beinhalten welche uns Interessieren
- Z.B: “The {Target} is on top of the {mask}“ wobei „{Target}“ ein Objekt aus unserem dictionary ist
- Wir nehmen die top 5 Ergebnisse und speichern diese ein

# Agenda

1. Aufgabenstellung

2. Verlauf

1. Wordnet

2. Wordvector

3. BERT

4. Evaluation

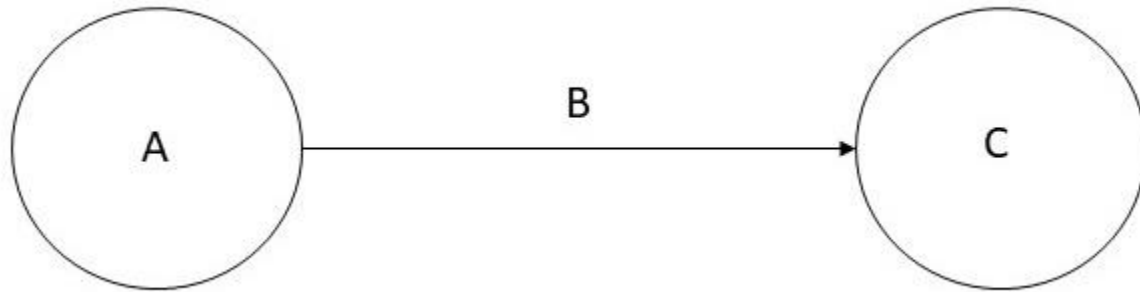
3. Final thoughts

4. Quellen



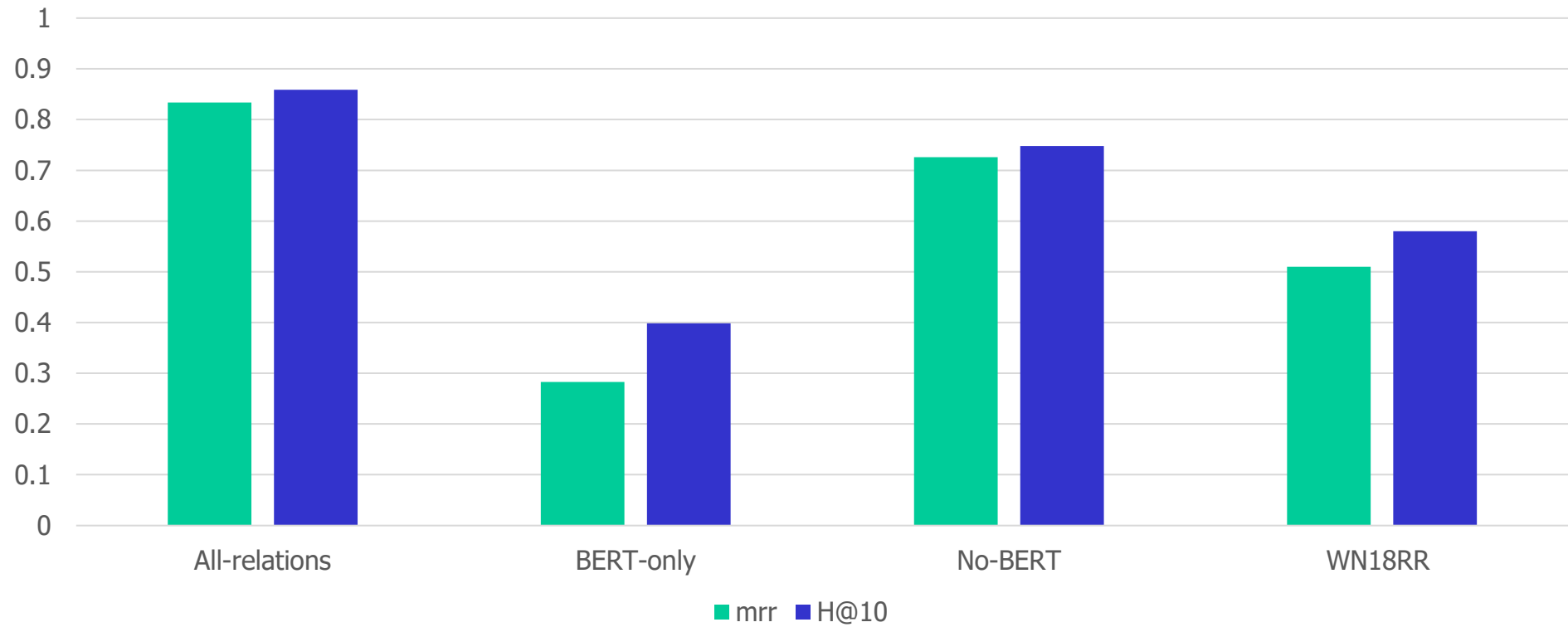
# Evaluation

- Subjekt und Objekt werden unabhängig voneinander verdeckt
- Modell rät Subjekt und Objekt
- Vergleiche jedes set miteinander
- Evaluiere MR, MMR und H@X



# Evaluation

Chart Title



# Agenda

1. Aufgabenstellung

2. Verlauf

1. Wordnet

2. Wordvector

3. BERT

4. Evaluation

3. Final thoughts

4. Quellen

# Final thoughts

## In Zukunft

- Evaluation verbessern mit besserem Modell
- FocusE: Betrachte Kantengewichte
- Datenstruktur neu anpassen
- BERT weiterhin verfolgen
- Relationen zu Verben erweitern
- Satzvervollständigung für Objekte

# Quellen

- neomodel: [https://neomodel.readthedocs.io/ /downloads/en/stable/pdf/](https://neomodel.readthedocs.io/en/stable/pdf/)
- neo4j: <https://neo4j.com/>
- NLTK/wordnet: <https://www.nltk.org/howto/wordnet.html>
- Ampligraph: <https://github.com/Accenture/AmpliGraph>
- BertMaskFlask: <https://gitlab.texttechnologylab.org/henlein/bertmaskflaskservice>

Vielen Dank für Ihre  
Aufmerksamkeit!