Outline

1. Topic Description
2. Collect
3. Prepare
4. Extension
5. Access
6. Future Workplan

1. Topic Description:
Relationships between Typical Nouns in Fairy Tales

1. Topic Description

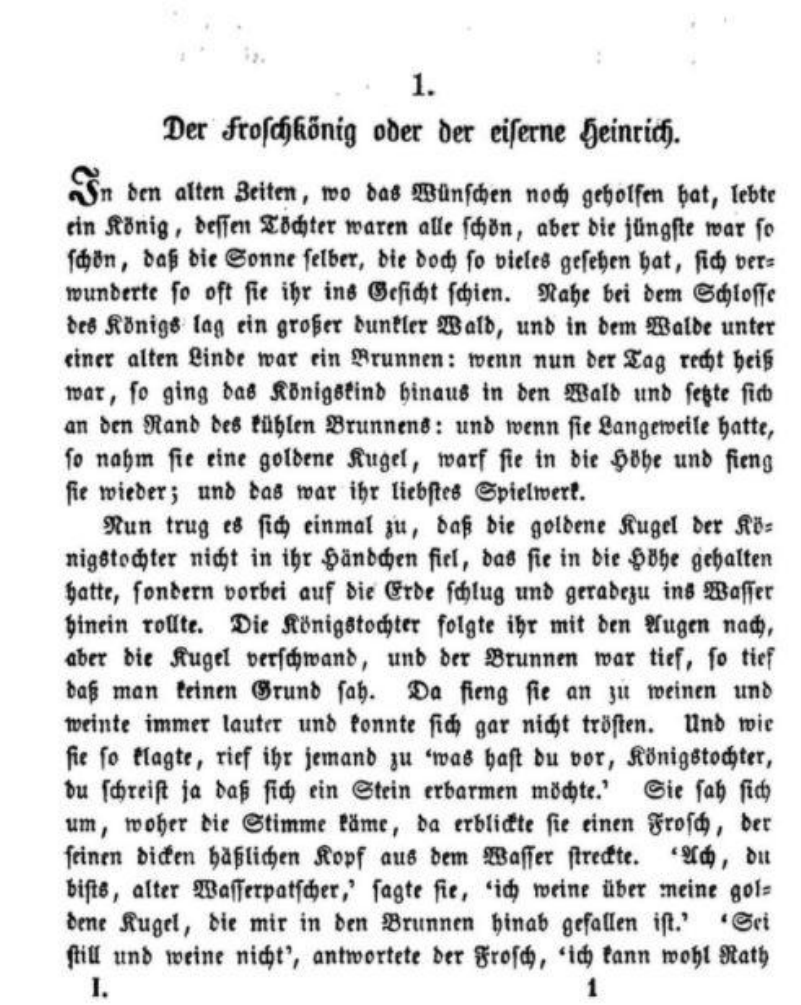
Relationships between typical nouns in Fairy Tales

- German Fairy Tales (Grimm brothers)
- Examine typical nouns (like king, witch etc.)
- Relationship between nouns (appearance in same sentence…)
- Represent typical nouns and relationships in a graph
- Query this graph and filter information

2. **Collect:**
Deutsches Textarchiv
Grimm Märchen

2. Collect

- Fairy Tales collected by Grimm Brothers
  - ➢ https://www.deutschestextarchiv.de/book/show/grimm_maerchen01_1857
- published 1857, 7th edition, 1st volume of 2
- digitized as a part of CLARIN-D
- CLARIN: "Common Language Resources and Technology Infrastructure"



1.

Der Froschkönig oder der eiserne Heinrich.

In den alten Zeiten, wo das Wünschen noch geholfen hat, lebte ein König, dessen Töchter waren alle schön, aber die jüngste war so schön, daß die Sonne selber, die doch so vieles gesehen hat, sich verwunderte so oft sie ihr ins Gesicht schien. Nahe bei dem Schlosse des Königs lag ein großer dunkler Wald, und in dem Walde unter einer alten Linde war ein Brunnen: wenn nun der Tag recht heiß war, so ging das Königskind hinaus in den Wald und setzte sich an den Rand des kühlen Brunnens: und wenn sie Langeweile hatte, so nahm sie eine goldene Kugel, warf sie in die Höhe und fieng sie wieder; und das war ihr liebstes Spielwerk.

Nun trug es sich einmal zu, daß die goldene Kugel der Königstochter nicht in ihr Händchen fiel, das sie in die Höhe gehalten hatte, sondern vorbei auf die Erde schlug und geradezu ins Wasser hinein rollte. Die Königstochter folgte ihr mit den Augen nach, aber die Kugel verschwand, und der Brunnen war tief, so tief daß man keinen Grund sah. Da fieng sie an zu weinen und weinte immer lauter und konnte sich gar nicht trösten. Und wie sie so klagte, rief ihr jemand zu 'was hast du vor, Königstochter, du schreist ja daß sich ein Stein erbarmen möchte.' Sie sah sich um, woher die Stimme käme, da erblickte sie einen Frosch, der seinen dicken häßlichen Kopf aus dem Wasser streckte. 'Ach, du bists, alter Wasserpatscher,' sagte sie, 'ich weine über meine goldene Kugel, die mir in den Brunnen hinab gefallen ist.' 'Sei still und weine nicht', antwortete der Frosch, 'ich kann wohl Rath


```xml
<?xml version="1.0" encoding="UTF-8"?>
<D-Spin xmlns="http://www.dspin.de/data" version="0.4">
  <TextCorpus xmlns="http://www.dspin.de/data/textcorpus" lang="de">
    <tokens>
      <token ID="w25570">1.</token>
      <token ID="w25571">Der</token>
      <token ID="w25572">Froschkönig</token>
      <token ID="w25573">oder</token>
      <token ID="w25574">der</token>
      <token ID="w25575">eiserne</token>
      <token ID="w25576">Heinrich</token>
      <token ID="w25577">.</token>
    </tokens>
    <sentences>
      <sentence ID="s10c4" tokenIDs="w25570 w25571 w25572 w25573 w25574 w25575 w25576 w25577"/>
    </sentences>
    <lemmas>
      <lemma tokenIDs="w25570">1.</lemma>
      <lemma tokenIDs="w25571">d</lemma>
      <lemma tokenIDs="w25572">Froschkönig</lemma>
      <lemma tokenIDs="w25573">oder</lemma>
      <lemma tokenIDs="w25574">d</lemma>
      <lemma tokenIDs="w25575">eisern</lemma>
      <lemma tokenIDs="w25576">Heinrich</lemma>
      <lemma tokenIDs="w25577">.</lemma>
    </lemmas>
    <tags>
      <tag tokenIDs="w25570">ADV</tag>
      <tag tokenIDs="w25571">ART</tag>
      <tag tokenIDs="w25572">NN</tag>
      <tag tokenIDs="w25573">KON</tag>
      <tag tokenIDs="w25574">ART</tag>
      <tag tokenIDs="w25575">ADJA</tag>
      <tag tokenIDs="w25576">NE</tag>
      <tag tokenIDs="w25577">$.</tag>
    </tags>
  </TextCorpus>
```

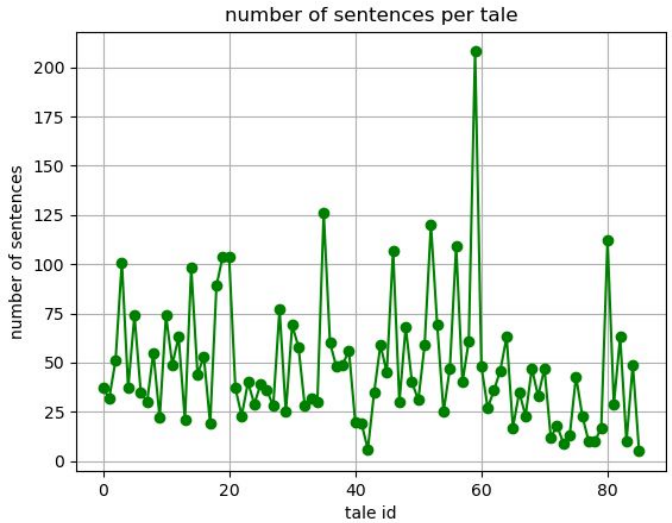
2. Collect

```
39
40  <tokens>
41      <token ID="w2d5b">funfzig</token>
42      <token ID="w2d5c">Thaler</token>
43  </tokens>
44  <corrections>
45      <correction tokenIDs="w2d5b" operation="replace">fünfzig</correction>
46      <correction tokenIDs="w2d5c" operation="replace">Taler</correction>
47  </corrections>
48
```

2. Collect

Corpus Statistics

	tales	tokens	lemma types	sentences
fairy tales only	86	147,659	33,532	4,125

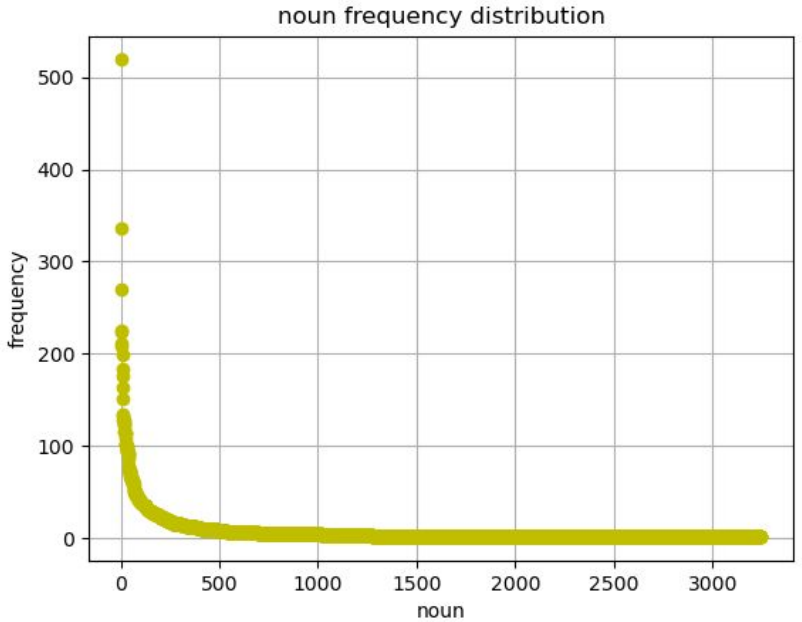


- 60. Die zwei Brüder: 208 sentences
- 86. Der Fuchs und die Gänse: 5 sentences
- mean: 47.97 sentences

2. Collect

Corpus Statistics

- number of nouns (NN + NE):
  - 22,256 tokens
  - 3,247 lemma types
- top 5 nouns:
  - ‘König’ *(king)* : 519
  - ‘Frau’ *(woman)* : 336
  - ‘Mann’ *(man)* : 270
  - ‘Kind’ *(child)* : 224
  - ‘Haus’ *(house)* : 211



3. **Prepare:** Modification of the XML document

3. Prepare

- Addition of annotation layers
- Modification of the xml document with Python ElementTree
  - API for parsing and creating xml data
  - represents the whole xml document as a tree
  - can be searched with XPath queries

3. Prepare

Additional Annotation Layers

- Tale boundaries
  - ➢ each tale starts with its title, e.g. “12. Rapunzel .”
  - ➢ collect all corresponding sentence ids from the start of a tale to its end (beginning of the next tale)

```
<tales>
    <tale ID="t11" title="12. Rapunzel ." sentenceIDs="s10cf s31c s31d s31e ..."/>
    <tale ID="t12" title="13. Die drei Männlein im Walde ." sentenceIDs="s10d0 ..."/>
</tales>
```

3. Prepare

Additional Annotation Layers

- Characters
  - manual selection of characters based on lemmas tagged as NN or NE
    - typical for tales, ideally high frequent

```
<characters>
    <character ID="c0" freq="519" taleIDs="t0 t2 ..." taleFreqs="9 7 ...">König</character>
    <character ID="c1" freq="94" taleIDs="t0 t5 ..." taleFreqs="6 9 ...">Königstochter</character>
</characters>
```

3. Prepare

Additional Annotation Layers

- Relations between the characters
  - simple assumption: if two characters appear in the same sentence, they share a relation
  - co-occurrence frequency as a measure for the strength of a relation

```
<relations>
    <relation ID="r1" freq="20" characterIDs="c0 c1" taleIDs="t5 t6 ..." taleFreqs="5 2 ...">('König', 'Königstochter')</relation>
</relations>
```

4. Extension:
Import XML to Neo4J

4. Extension

Neo4j

- a native graph database
- stores connections between data
- uses the Cypher language
  - ➢ graph-optimized query language

4. Extension

Advantages of Neo4j for our project

- Neo4J models relationships between nodes
  - ➢ we want to investigate relationships between nouns
- the graph can be searched efficiently with relatively short queries
- data can be visualized natively as a graph
- data can be easily explored through a graphical interface

4. Extension

Import XML to Neo4J: Example of setting up the graph structure


```
CALL apoc.load.xml("file:///tales_neo4j.tcf.xml",'/TextCorpus/tokens')
YIELD value as tokens
UNWIND tokens._children AS token
CREATE (t:Token {id:token.ID, text:token._text})
WITH collect(t) AS tokens
UNWIND apoc.coll.pairs(tokens)[0..-1] AS value
WITH value[0] AS a, value[1] AS b
CREATE (a)-[:NEXT_TOKEN]→(b);
```

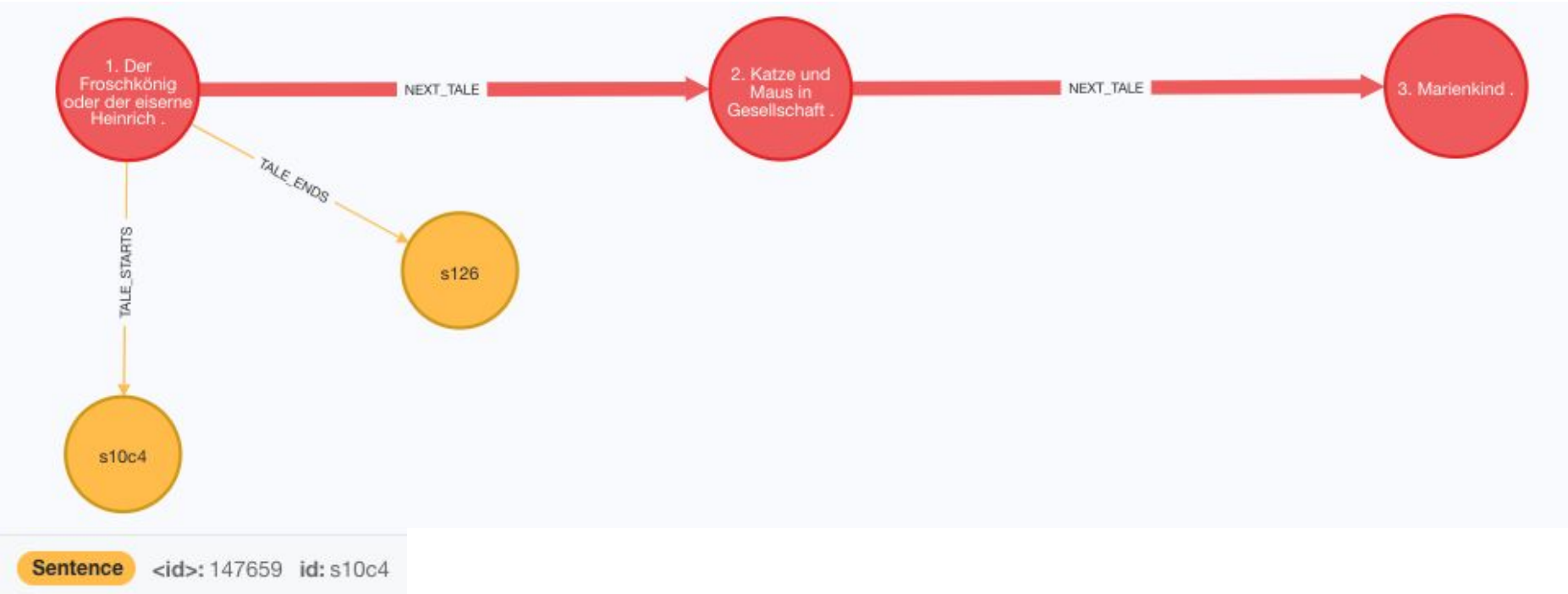
5. Access:
Corpus as a Graph in Neo4J

5. Access

Tales

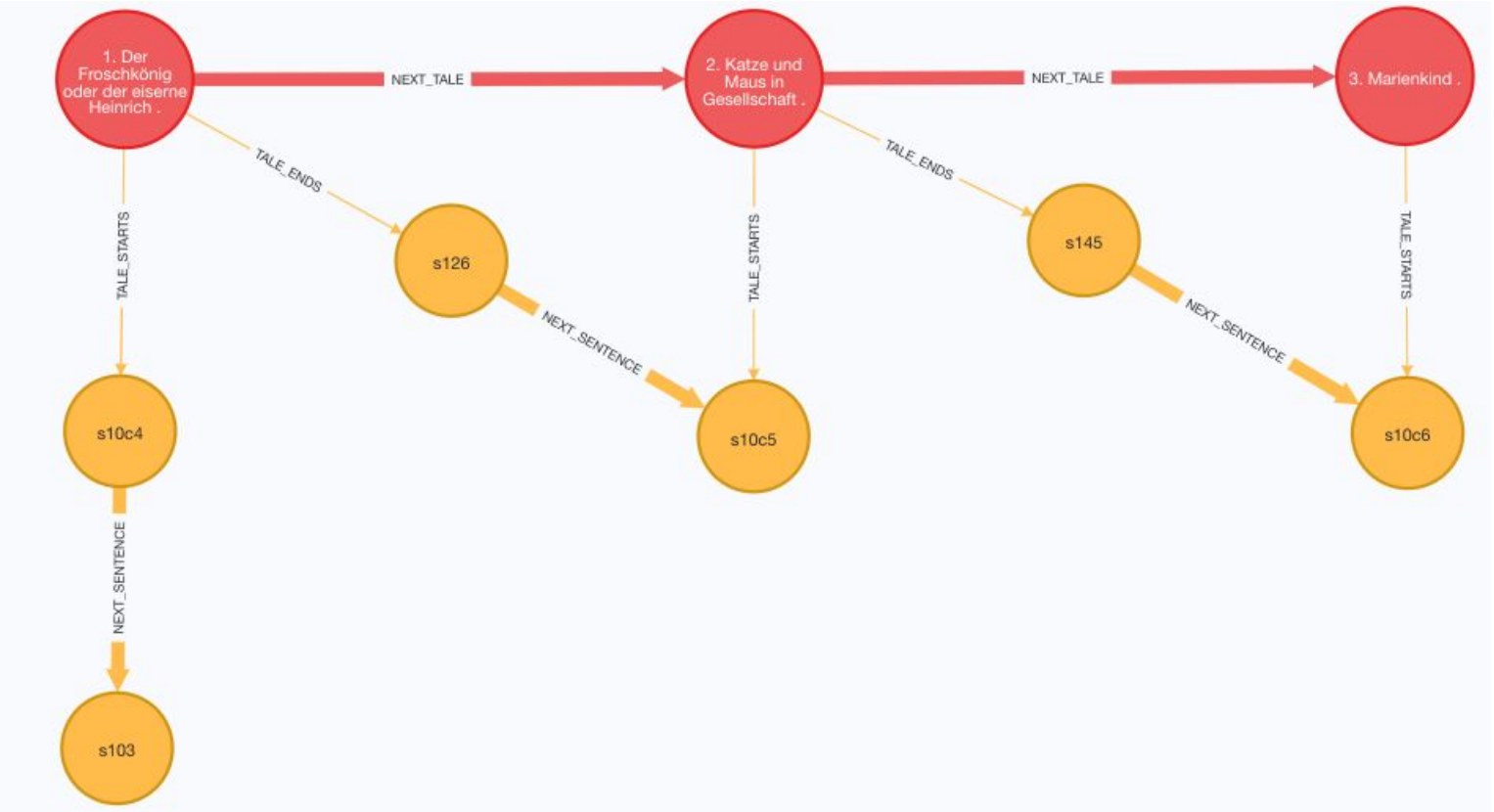
5. Access

Sentences



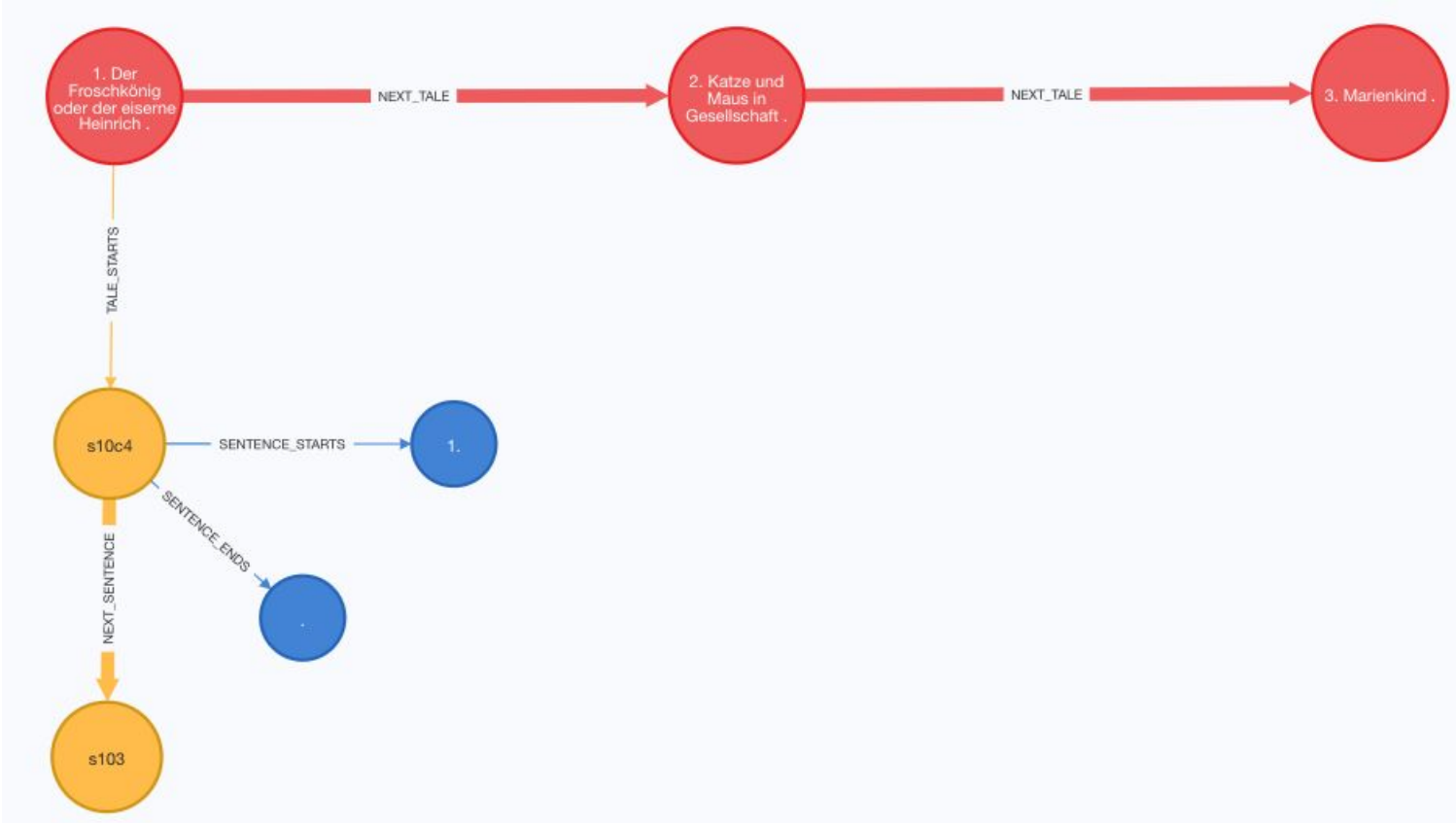
5. Access

Sentences



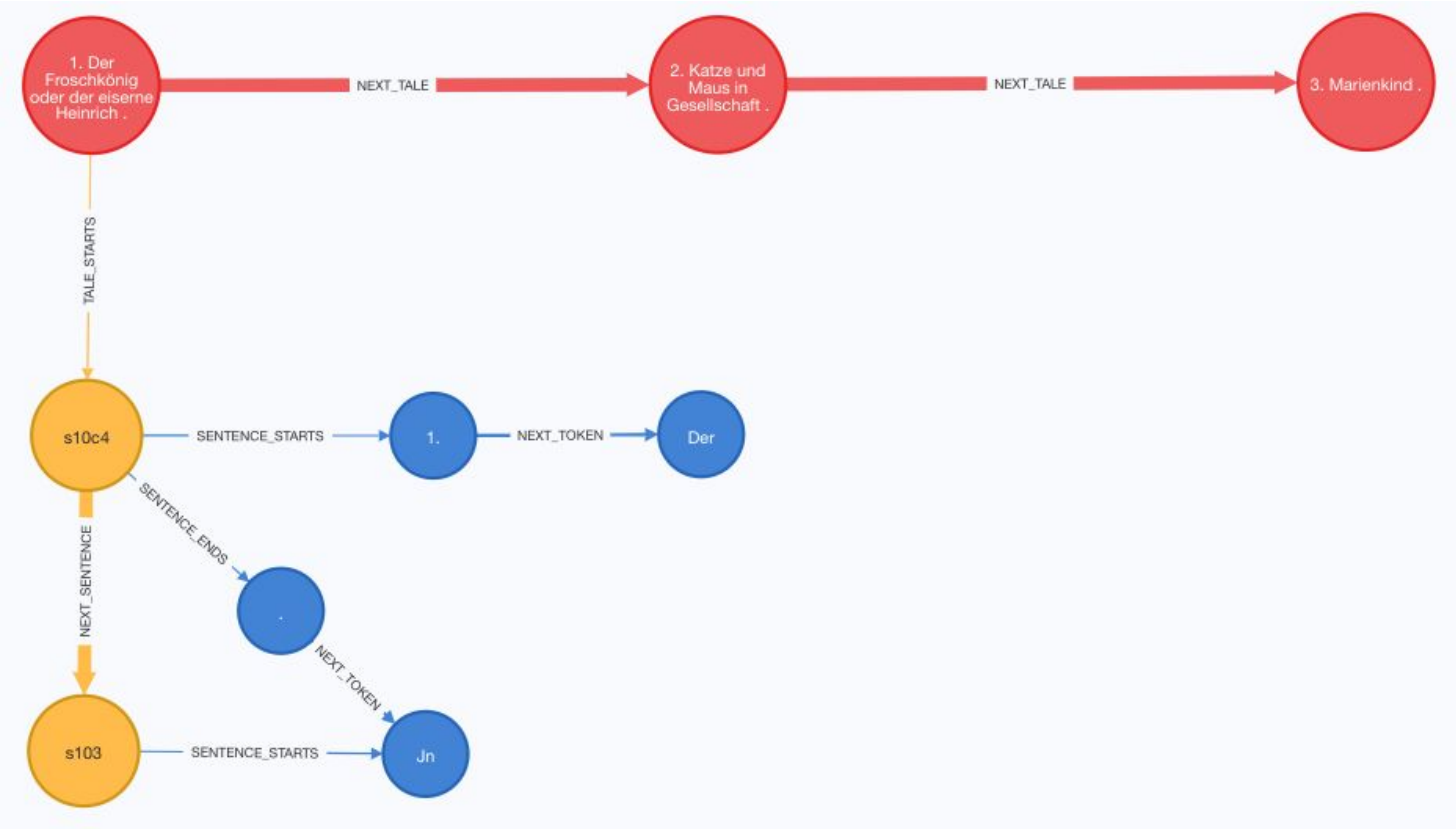
5. Access

Tokens



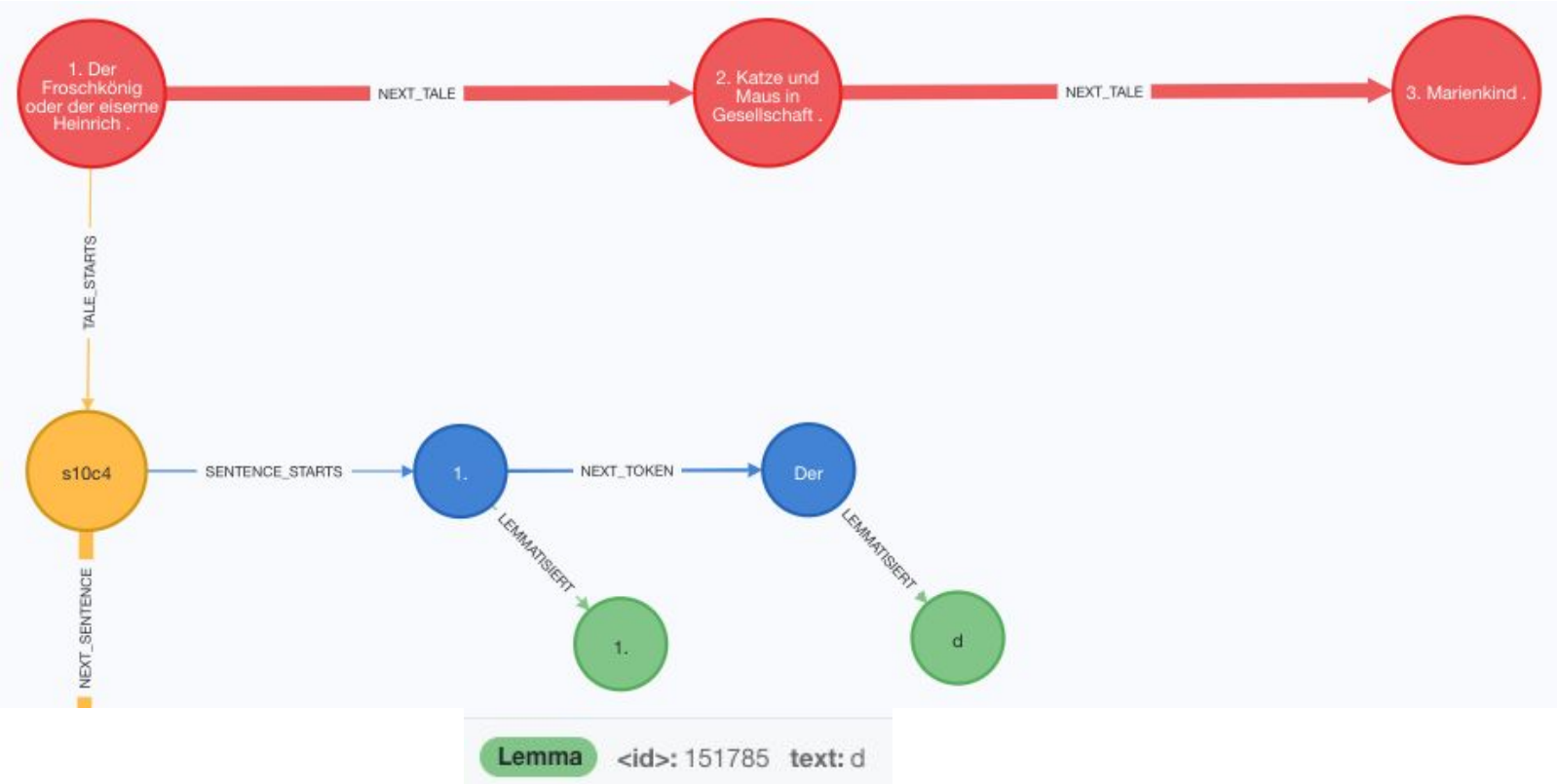
5. Access

Tokens



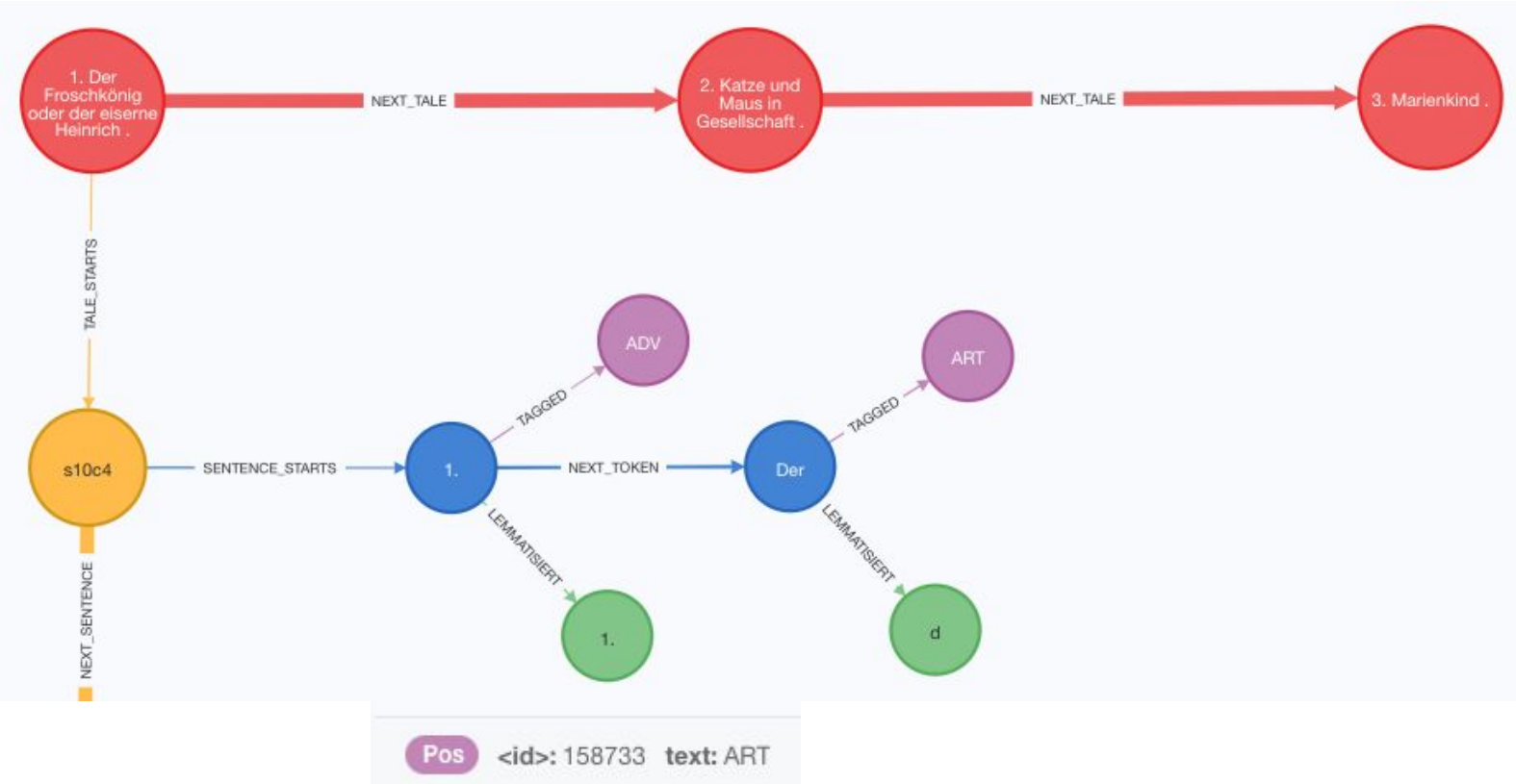
5. Access

Lemmas



5. Access

POS Tags

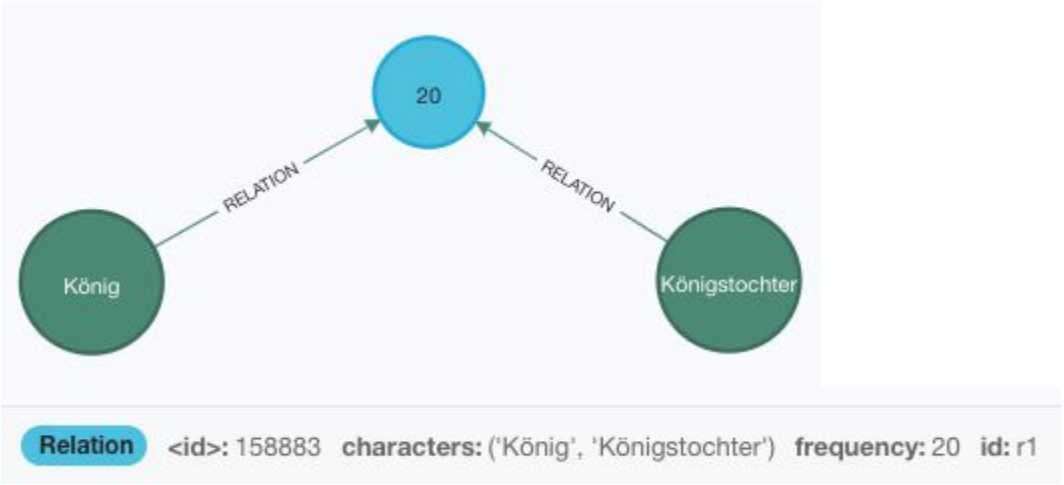


5. Access

Characters

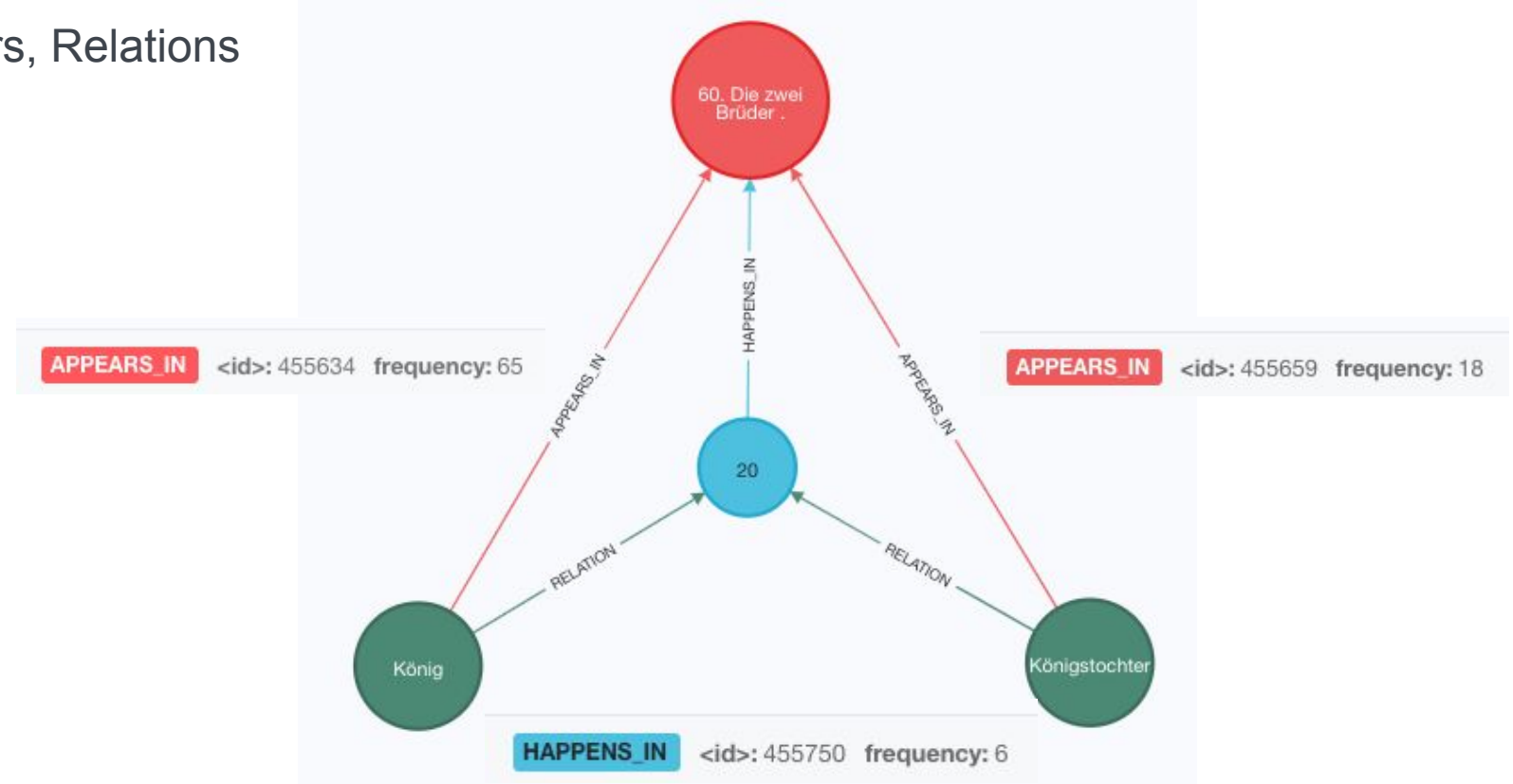
5. Access

Relations

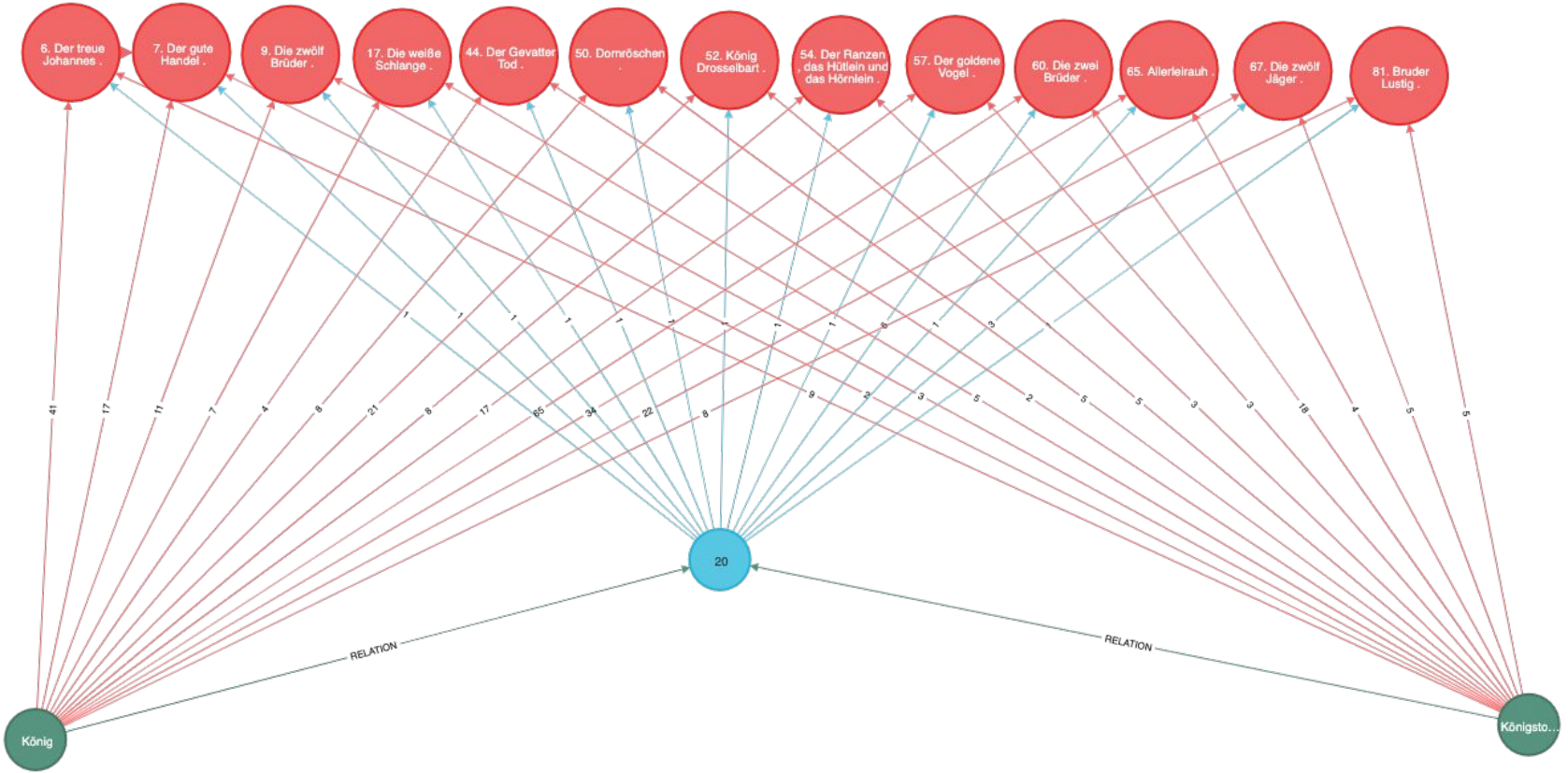


5. Access

Characters, Relations



5. Access


6. Der treue Johannes .
7. Der gute Handel .
9. Die zwölf Brüder .
17. Die weiße Schlange .
44. Der Gevatter Tod .
50. Dornröschen .
52. König Drosselbart .
54. Der Ranzen , das Hütlein und das Hörnlein .
57. Der goldene Vogel .
60. Die zwei Brüder .
65. Allerleirauh .
67. Die zwölf Jäger .
81. Bruder Lustig .
20
König
Königsto...
RELATION
RELATION

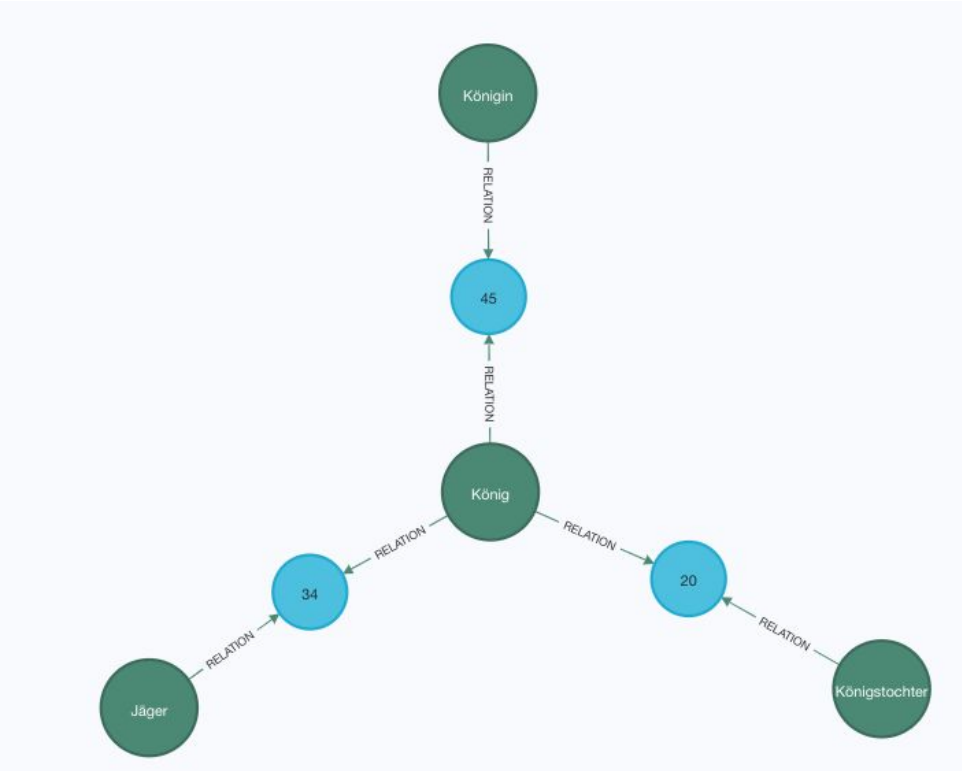
5. Access

Example Query: find all relations of *‘König’* that have a frequency >= 20

```
MATCH (c1:Character {name:'König'})-[:RELATION]→(re:Relation)←[:RELATION]-(c2:Character)
WHERE toInteger(re.frequency)⩾20
RETURN c1, c2, re
```

5. Access

Example Query: find all relations of *‘König’* that have a frequency >= 20



6. Future Workplan

6. Future Workplan

- Explore properties of nouns and their relations to other nouns
- Explore more layers of relationships:
  - occurrence in same tale but not same sentence
  - occurrence in consecutive sentences
  - distance in occurrence by number of sentences in between

6. Future Workplan

- Investigate specific tales
- Possibly find a better rule to automatically annotate relations
  - ➢ pronouns, synonyms, etc. are not taken into account
- Use Pointwise Mutual Information instead of frequency
- Expand research to the 2nd volume

Sources

- Deutsches Textarchiv - https://www.deutschestextarchiv.de
  - Grimm Fairy Tales (1857): https://www.deutschestextarchiv.de/book/show/grimm_maerchen01_1857
- neo4j.com - https://neo4j.com
- extension: https://neo4j.com/labs/apoc/4.2/import/xml/
- our Github repository: https://github.com/Thommy96/tcf-tales

Thank you for your attention!

We are happy to answer questions.