2. ITA Projekttreffen: John Ziegler

An Investigation of the Bible

Felix, Johannes, Sven

Agenda

1. current progress

2. open points

3. Test, pipeline and commentary

4. co-reference resolution

Current progress

1. Keyword Extraction

2. Character detection

3. Object creation

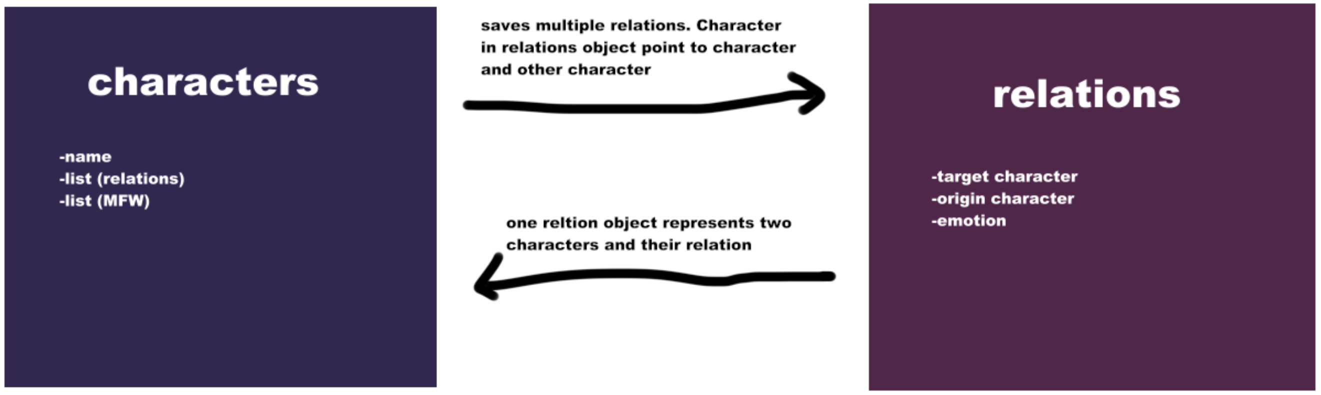
4. Emotion detection & graph generation

Keyword extraction

- used to be the topic of Aileen
- currently, keywords each character are extracted and added to character object
- commented and running so fare
- However, keywords in graph detection are mocked so fare

Character detection

- spacy entity recognizer (label: Person)
- rule based matching mit Character patterns.
- Extracted ID‘s from patterns using Wikidata
- Matching via knowledge base



characters

-name
-list (relations)
-list (MFW)

saves multiple relations. Character in relations object point to character and other character

relations

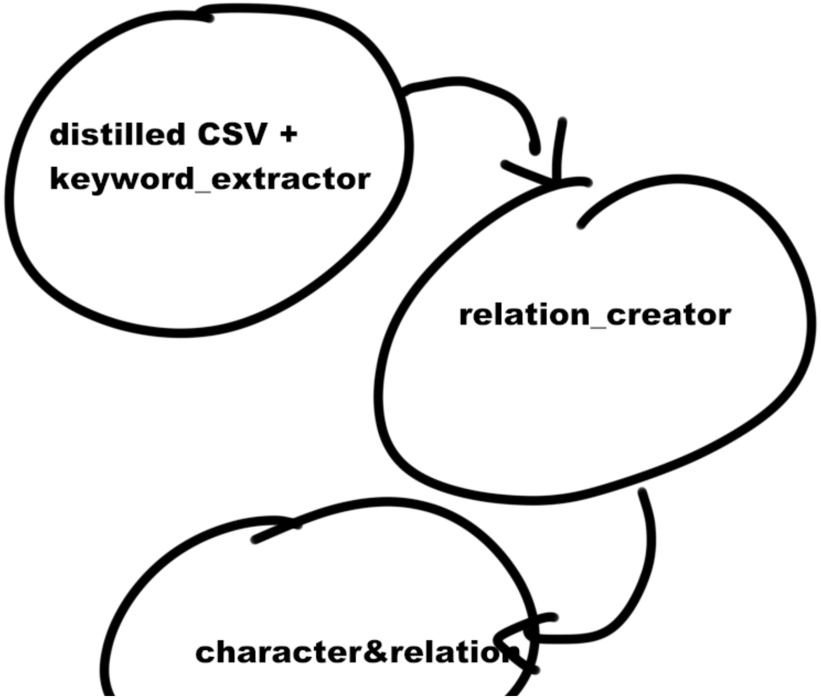
-target character
-origin character
-emotion

one reltion object represents two characters and their relation

Problems and Tasks:

How do we save both Classes?
How do we create them?
How do we deal with the Issue of having multiple characters with the same name?
How do we deal with multiple rows representing the same relationship?
How do we deal with implementing this step in our pipeline?

Pipeline:



distilled CSV + keyword_extractor

relation_creator

character&relation

Object creation

- Create Python objects, which were created from information in the character detection process.(distilled dataframe from graph detection)

- Character object contains information about relations with other characters and character specific data from the keyword extraction process.

- A relation object represent that two characters have a relationship and have a numeric value representing a good/bad relationship.

- Objects are python objects, so to further use them in other programms we use the Pickel module.

- This serves intermediate step in our Pipeline. It is important to have all our information ready to be loaded by other steps in our Pipeline

Emotion detection

- Bayesian classifier: TextBlob
- Bag of Words: 600 positive, 448 negative
- Both create value, each row both scores are processed by:

```
if score > 0.75:
    Emotion = 1.0 → Positive
elif score < -0.75:
    Emotion = -1.0 → Negative
else:
    Emotion = 0.0 → Neutral
```

- Code is commented but needs more detail

Graph generation

Base

- distill dataframe to distinct set of relations
- aggregation follows emotion detection
- create edges from characters
- color edge based on emotion score
- Code is commented but needs more detail

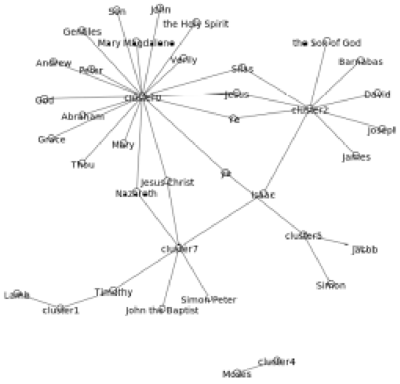
Graph generation

Cluster

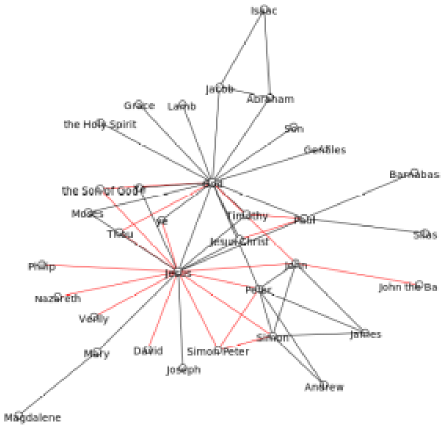
- cluster based on keywords
- keywords in word2vec
- cluster all keywords in N centroid using kmeans
- [„Death“, „Hell“, „Love“] → [1,1,4]
- Person is in all clusters >= Threshold → Threshold = 2 → cluster 1 für person with keywords
- find cluster in graph by Depth-First-Search algorithm
- apply cluster in graph
- Summarize N clusters at one node
- aggregate emotion from cluster to other nodes

Graph generation

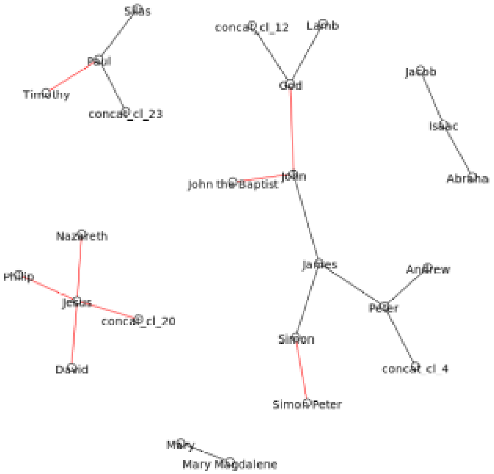
Cluster aus den Keywords je person



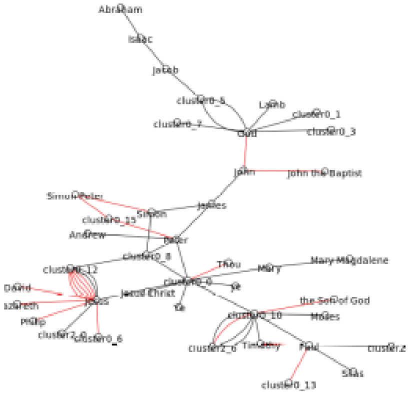
Beziehungsgraph



n cluster an einer node zusammenziehen



Graph (←) und Cluster (↑) „verschränkt“



Open Points

Graph generation

- after clustering a person, this person can not be added to another cluster
- adjust node size of cluster by number of people in cluster
- find family relations

Character detection

- Text
-

Test, pipeline and commentary

Tests

- Originally Aileens Topic
- First tests are written by Aileen, has to be taken up by one of us

Pipeline

- currently running python programms one-by-one

Commentary

- Code is mostly commented but needs finish at parameters, return and function definition