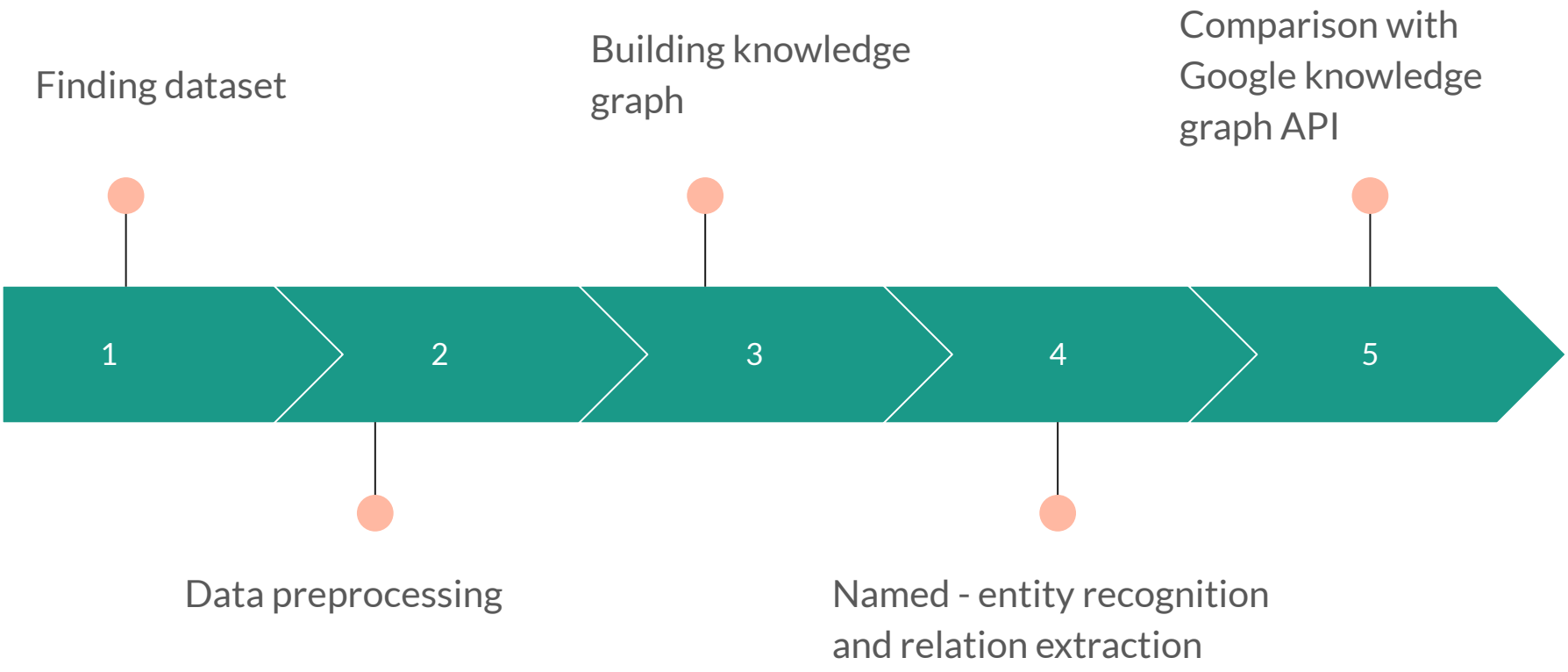
Unstructured Text to Knowledge Graphs: Using NLP tools and Google KG APIs

Team : Anusha Manur, Nandita Srinivasan, Urmil Parikh

Objective

Develop an end-to-end solution for building a knowledge graph from text using NLP tools and interfacing with the knowledge graph using Google's Knowledge Graph APIs

Project overview



Dataset

IMDB 5000 Movie Dataset

- 5000+ movie data scraped from IMDB website
- We take a subset of features of the IMDB dataset

Dataset

- Our dataset contains details of 5000 movies with the following features
  - Movie title , Genre , Director, Actors (3) , Language

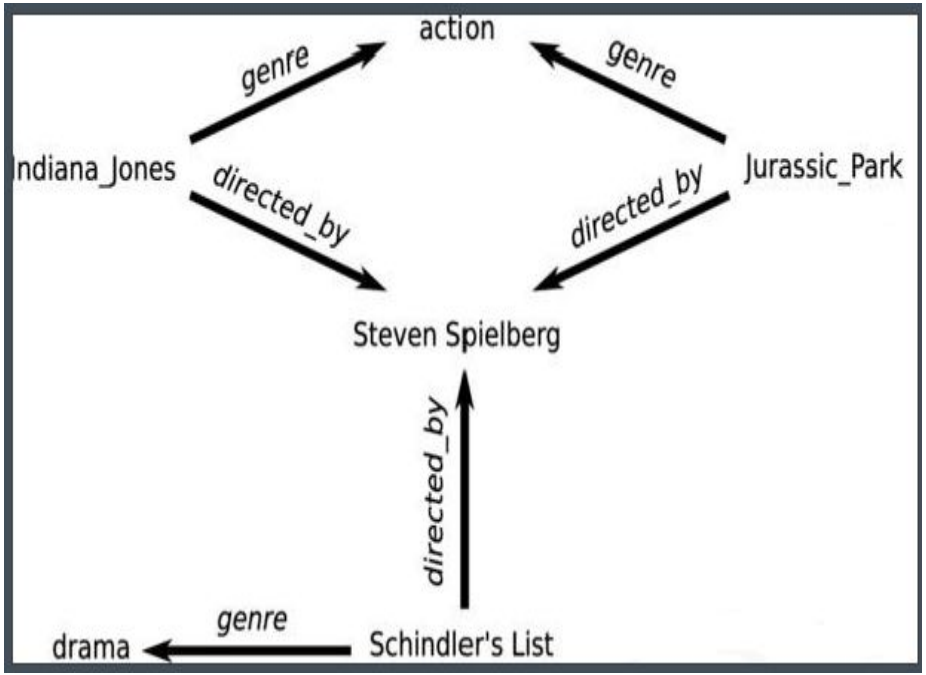
	director_name	actor_2_name	genres	actor_1_name	movie_title	actor_3_name	language
2	James Cameron	Joel David Moore	Action|Adventure	CCH Pounder	Avatar	Wes Studi	English
3	Gore Verbinski	Orlando Bloom	Action|Adventure	Johnny Depp	Pirates of the Ca	Jack Davenport	English
4	Sam Mendes	Rory Kinnear	Action|Adventure	Christoph Waltz	Spectre	Stephanie Sigma	English
5	Christopher Nola	Christian Bale	Action|Thriller	Tom Hardy	The Dark Knight	Joseph Gordon-L	English
6	Doug Walker	Rob Walker	Documentary	Doug Walker	Star Wars: Episode VII - The Force Awakens		
7	Andrew Stanton	Samantha Morto	Action|Adventure	Daryl Sabara	John Carter	Polly Walker	English
8	Sam Raimi	James Franco	Action|Adventure	J.K. Simmons	Spider-Man 3	Kirsten Dunst	English
9	Nathan Greno	Donna Murphy	Adventure|Anima	Brad Garrett	Tangled	M.C. Gainey	English

Data preprocessing

- The most relevant movie details are considered.
- Missing values are handled.
- Duplicates are removed.
- Unicode characters removed.
- Special characters are removed.

Building a knowledge graph

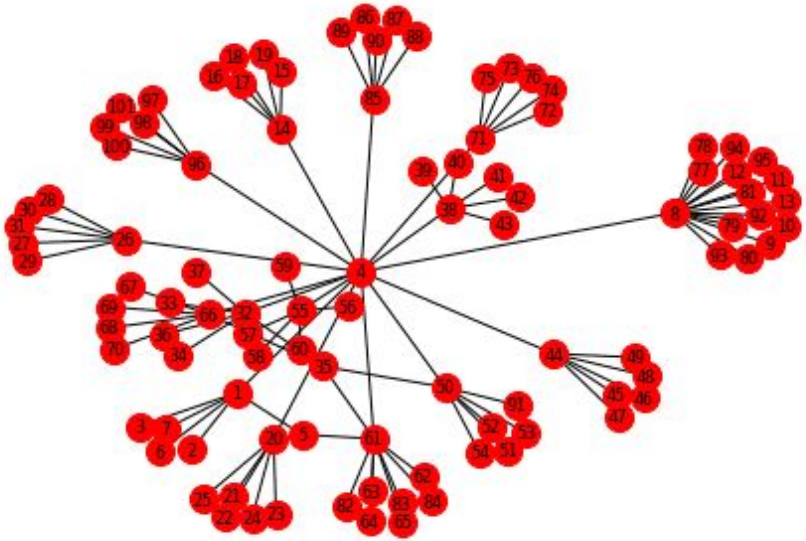
- Each feature of the dataset is a node in the knowledge graph and the relation between the two features is represented by the edge between the two nodes.
- Construction of a graph allows nodes to multiple relations and enables easy querying.



Credits - IOS press content library

Building a knowledge graph

- We parse the dataset to build a graph using NetworkX library.
- The image shows the graph constructed for 20 , 30 and 40 movies where each node has a unique node ID.



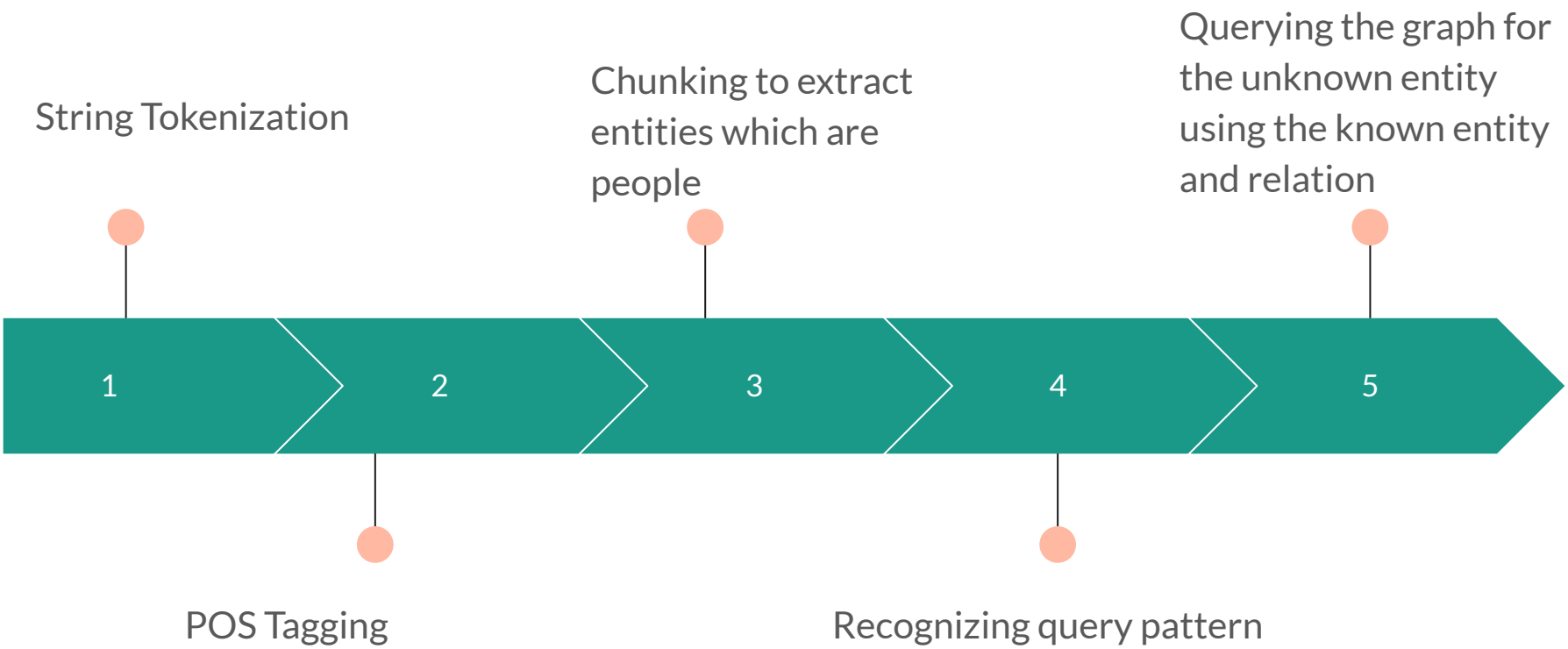
Querying the knowledge graph

Overview

- Creation of query dataset
- Named entity recognition
- Relation extraction
- Query the knowledge graph
- Compare with Google knowledge graph API

Creation of query dataset

Project overview



Tokenization and POS - tagging

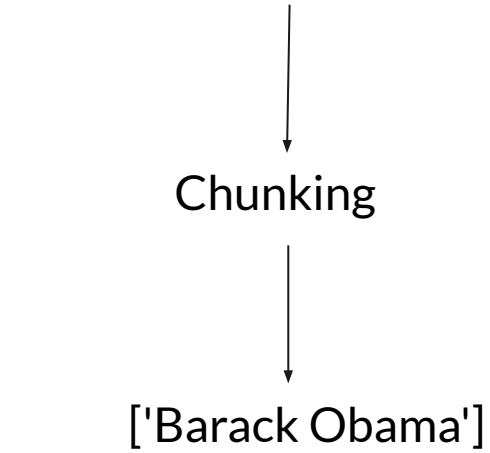
- Each query is first tokenized.
- We then POS-tag (parts of speech) the tokens.
- This allows us to establish a pattern for how to recognize the required entities based on context.

Chunking

- Chunking is done to extract the complete names of the actors/directors (instead of 2-3 words) as given in the knowledge graph
- This is done to extract the complete name as a single entity.

Example :

"Barack Obama was the president of our country."



Recognizing query pattern

- We consider various tenses of nouns, pronouns, verbs and determiners for entity extraction.
- We check with a predefined list of relations and the respective synonyms to extract relations.

Example :
"Steven Spielberg directed the movie Jurassic Park"

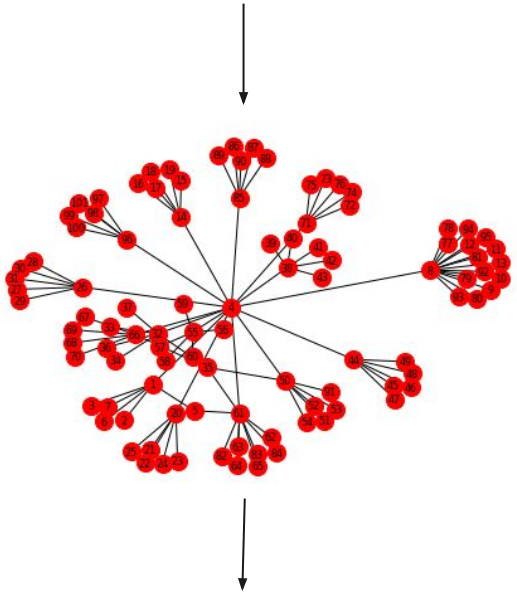
Recognizing query pattern

“Steven Spielberg directed the movie Jurassic Park”

Querying the graph

- From the function, we get a tuple of the form [Entity, Entity_Name, Relation]
- We then query the graph with the entity. That is, we find a node of the type “Entity”, whose value is the entity name, and follow the edge attributed with the relation to find the required nodes.
- For example, if the question is, “Who stars in I am legend?” We find the “title” node(entity) whose value is “I am Legend”(entity_name) and follow the edge that goes to the “Actor”(relation=’acts’) node.

Who stars in I am Legend



[‘will smith’,’willow smith’,’alice braga’]

Testing the NER function

- The queries dataset contains some possible questions regarding the movies
- Examples questions are - “Who are the actors in Iron Man 2?” or “Who is the director of Casino Royale?
- Each of the queries are fed to a function to get the corresponding [entity,entity_name,relation] tuple
- The graph is queried with the tuple, and the obtained result is stored in a list.
- Our assumption is that, if the graph does not return an answer for an entity, then the NER process was not successful, and hence the model is given a score of 0 for that entity.

Querying the Google Knowledge graph

- The queries dataset contains some possible questions.
- Since we cannot use the entire sentence to query the Google knowledge graph and only an entity, we extract the main entity from the question and use it to query the Google Knowledge Graph.
- We compare the results obtained from the Google knowledge graph with the generated graph.

Results

- Our knowledge graph detects entities and relations and returns correct answers to 61% of the queries.
- The output of Google knowledge graph is as shown below

```
api_key = API_KEY
service_url = 'https://kgsearch.googleapis.com/v1/entities:search'

query = 'i am legend'
params = {
    'query': query,
    'limit': 4,
    'indent': True,
    'key': api_key,
}
url = service_url + '?' + urllib.urlencode(params)
response = json.loads(urllib.urlopen(url).read())
mov_details=[]
names=[]
for element in response['itemListElement']:
  if 'detailedDescription' in element['result'] and 'name' in element['result']:
      print (element['result']['name'],"--", element['result']['detailedDescription'])
```

```
u'I Am Legend is a 2007 American post-apocalyptic science fiction horror film based on the novel of the same name, directed by Francis Lawrence ar
 u'I Am Legend is a 1954 science fiction horror novel by American writer Richard Matheson. It was influential in the development of the zombie-var
dy': u'I Am Legend is a South Korean television series starring Kim Jung-eun that was broadcast on SBS on August 2 to September 21, 2010.', u'lice
```

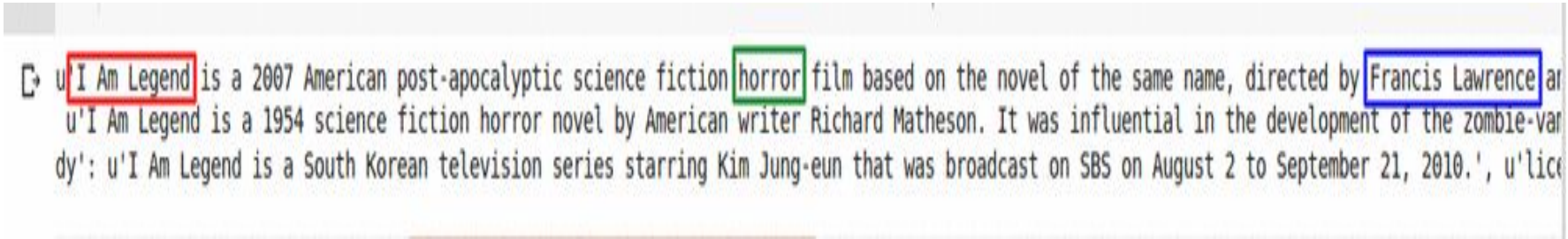
Results

Results from our knowledge graph

```
entity_tuple=create_entity_tuple("Who directed i am legend?")
print(entity_tuple)
answer_list=get_entity_relation_from_graph(entity_tuple[0],entity_tuple[1],entity_tuple[2])
print(answer_list)
```

```
('title', 'i am legend', 'director')
['francis lawrence']
```

Results from the Google knowledge graph



Assumptions

- No pre-trained model like spacy or StanfordNLP has been used for named entity recognition and relation extraction.
- We have tested our model on the queries we manually generated.
- To perform NER, we define a pattern-based context for the queries in the generated dataset. Hence, the queries we use to search the graph should conform to the same.

Conclusion

- Our NER function and knowledge graph perform similarly to that of Google's Knowledge Graph API, with respect to the context of Movies.
- NER and knowledge graph creation can have diverse applications in being able to classify and summarize content, creating effective search engines and also in chat based applications.
- Knowledge graphs prove to be very powerful in offering semantic context to otherwise unstructured, natural language text.

References

[1] Code for Continuous Chunking: https://stackoverflow.com/questions/48660547/how-can-i-extract-gpelocation-using-nltk-ne-chunk

[2] Google Knowledge Graph Search API https://developers.google.com/knowledge-graph/