Background

A knowledge graph is a collection of data that represents the real-world relationships between entities. These (named) entities can be anything, such as people, places, things, or ideas.

The data in a knowledge graph is usually organized into a graph structure, with each entity represented as a node and the relationships between entities represented as edges. The goal of a knowledge graph is to provide a way to represent real-world data in a way that is easy for computers to understand and reason about. Knowledge graphs have become increasingly important in recent years as they have been used to power a variety of applications, such as search engines, recommendation systems, and question answering systems.

A knowledge graph is populated through a process of data collection and data integration. This process begins with the identification of sources of data that can be used to populate the knowledge graph. Once these sources have been identified, the data is collected from these sources using information extraction techniques and integrated into the knowledge graph.

Examples of Knowledge Graphs are Google Knowledge Graph[1] or Wikidata[2], which are constructed and maintained mainly through user-supervision. There are a number of reasons why knowledge graphs are often manually constructed. One reason is that it can be difficult to automatically extract all of the relevant information from unstructured data sources. Some aspects involved are:

- Identifying the source(s) of the relevant information
- Determining what information is relevant to the query later on
- Extracting the information from the source(s)
- Organizing and presenting the information in a way that is useful to the user

The advantages of manually constructing a knowledge graph can help to ensure its accuracy and completeness, while disadvantages can be lack of standardization and consistency. Despite there being much value in the use of such collections of Knowledge, the creation of a knowledge graph through manual means is time-consuming and expensive. The limitations of this automated extraction also limits the speed at which this technology is spread, and there is a considerable need for more automated Knowledge Graph construction methods.

Data description

In this assignment we will undertake the general challenge of extracting information from 2 different sources:

1. An unsupervised (albeit structured) large corpus: the entirety of Wikipedia.
2. Event registration and documentation for nuclear powerplants in the US

Wikipedia

A Wikipedia dump[3] is a copy of all of the content from Wikipedia. This includes all of the articles, images, and other media files. We provide a somewhat stripped/cleaned version of Wikipedia which

[1] https://en.wikipedia.org/wiki/Google_Knowledge_Graph
[2] https://www.wikidata.org/wiki/Wikidata:Introduction
[3] https://dumps.wikimedia.org/backup-index.html

saves you a considerable amount of computing power required to start your project. This stripped version has had tables, headers and other "non-text" removed[4].

Technical language data

Language data from industry often poses unique challenges due to its specialized nature and domain-specific terminology. Unlike generic text found on the internet, industry-specific language is often rife with technical jargon, abbreviations, and context-dependent meanings. This requires a deep understanding of the specific field to accurately process and interpret the data.

To be familiarize with this challenge, a second dataset is provided which describes a collection of events regarding unexpected reactor trips at commercial nuclear power plants in the U.S[5]. It contains a selection of metadata, a short-hand description as well as a longer abstract describing the occurrences in detail.

Assignment

The overall goal consists of the following perform information extraction on these dataset. In the end, you should end with a collection of triplets (**[subject, relation, object]**) which could populate a Knowledge Graph.

Perhaps for a particular use-case or subdomain in these datasets.

- Identify and label entities and relations of interests
- Select appropriate data sources that (likely) contain that information
- Build working models that can extract this information
- Evaluate the performance of extraction in parts and as a whole

Example

We will demonstrate the task at hand below on the following text and as a toy example in order to demonstrate what you will need to do (albeit more numerous and more systematic) for the assignment. We designate interest in people, their personal characteristics and their involvement in a political party. For the assignment, you are expected to come up with your own "subdomain" and relevant entities and relations.

> Barack Hussein Obama II (born August 4, 1961) is an American politician who served as the 44th president of the United States from 2009 to 2017. A member of the Democratic Party, he was the first African-American president of the United States. Obama previously served as a U.S. senator from Illinois from 2005 to 2008 and as an Illinois state senator from 1997 to 2004.

1. **Identify and label entities and relations of interests**

[4] https://github.com/attardi/wikiextractor
[5] https://nrcoe.inl.gov/InitEvent/

First, we should identify what entities and relations are interested in. Next, we need to define annotation guidelines in order to have data labelled by multiple annotators. For this toy example, we will focus on the following:

Entities: Persons, Organizations and Dates, and

Relations: *born_in*, member_of

Now, rather than provide preset annotation guidelines, we will ask several questions that you may need to clarify when writing your own:

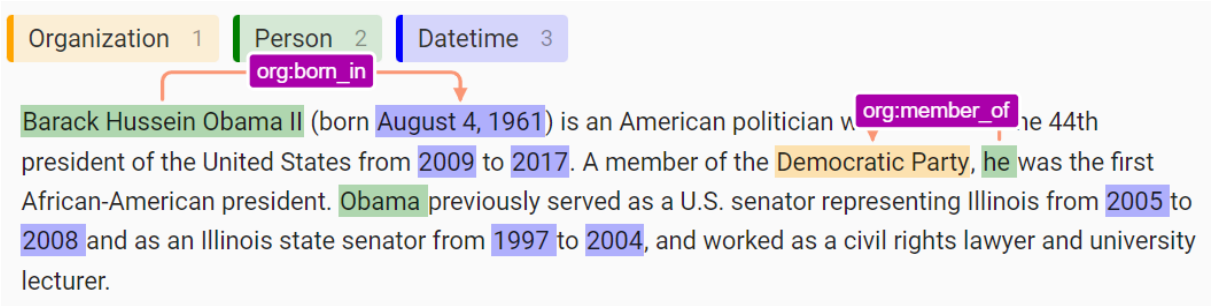
Entities:

- Persons: Do fictional characters count? Do they need to be alive right now? What about pronoun references such as he/she/they/them?
- Dates: Do you need day, month and year for it to qualify? What if it also has a time? What if day and/or month is missing?

Relation:

- Which way does the relation go?
- How explicit does the information need to be?

In our toy example we come to the following labeling:



2. <u>Select appropriate data sources that (likely) contain that information</u>

[This is essentially open to your group but you need to make a well-justified choice and deliberate pros and cons. We will discuss several options and some pro's and con's.

1. we could filter on pages that contain "(person)" in the title, but this seems to be only the case when the entity at hand requires disambiguation. It means it will be fairly correct, but it will also miss a lot of relevant sources.
2. We could filter on certain words appearing in the first paragraph. E.g. if there an occurrence of ("born [DATE]"), you can come to the conclusion it is likely to be about a person. Adding another filter for ("is a politician") allows you to filter for politically involved people. This will work well if 2 assumptions are fulfilled: 1) the inform is actually there and 2) the person has their **known** birthdate/occupation given in this specific structure. On certain pages, there actually may variations. For example: [Barack Hussein Obama II … is an American politician] has a grammatical variation based on additional information (that hes an American politician).
3. Use a topic-model approach using Bag-of-Words or TF-IDF to search for relevant terms.

3. Build working models that can extract this information

Next, you are asked to build models that can extract this information. Based on our example above, our extraction should lead to two triplets:

[Barack Hussein Obama II, born_in, 04-08-1961]

[Barack Hussein Obama II, member_of, Democratic Party]

Tutorial 3 will cover this process more in detail, but here we'll discuss it on a high level.

1. First, we will classify the entities in the text (the process of NER)
2. Second, we will classify relations over the entities

4. Evaluate the performance of extraction in parts and as a whole

By now, you should have already used multiple NLP-methods in order to process, select and classify text from the provided dataset. Largely, an important component on the assignment is the verification.

1. How well does your information selection process work?
2. How did your labeling process go?
    a. What are your annotation guidelines?
    b. How much did you (dis)agree?
    c. How did you resolve conflicts?
3. How well does your (trained) IE-pipeline work?
4. What are the effects on the evaluation dimensions (e.g. accuracy) when putting all of these component in sequence in order to extract information?

Wikipedia subdomain

For the Wikipedia dataset, you are free to see out desired subsets of entities and relations of your interest. E.g. if you are interested in "athletes", you'll need to come up with relevant entities and relations for this domain and develop a strategy to filter athlete-related articles from the raw pile of data.

Technical data label-space

As said earlier, technical data often requires intensive collaboration with domain experts in order to determine task purpose, entities, relations, etc. Due to lack of available nuclear engineering experts to assist you with this assignment for this course, we will provide suggestions for your entities/relations labeling scheme but leave the interpretation and associated annotation guidelines to you. You are allowed to add/remove/modify entities/relations as you see fit.

Entities

- Datetime
- Location
- Event

    - Unexpected event
    - Expected event
- Cause
- Attribute
- Activity

Relations

- has_part
- has_participant
- has_activity
- caused_by
- happened_at
- happened_on

End result

The end result from your work will be the extracted triplets ([subject, predicate, object]) and all the work in order to extract those. Making these results queryable in an actual Graph Database will reward a bonus, but the focus should lie on everything before it.

Resources

We highly encourage you to use environments like Google Colab in order to do classifications which are GPU-based, and to write and/or use efficient and highly optimized code which makes maximum use of limited resources (e.g. multiprocessing, preprocessing, etc.).