

# Information Retrieval – Mini Project

## Goal

The overall goal is to create a basic QA system that can retrieve single resources that answer a given question.

## Proposed workflow

1. Represent an RDF resource by its concise bound description (CBD)
2. Train transformer to generate embedding of resource after
  - a. reading CBD of said resource or
  - b. QALD question from training set
3. Ensure that resources and literals are represented correctly (text embeddings vs. KG embeddings)
4. Evaluate the approach on test set of QALD 9

For the sake of rapid training amount, we limit the resources by their type.

### 1. Representation of a single resource

A resource  $r_i$  within a knowledge graph can be represented by its concise bound description (CBD). For each resource  $r_i$  of the chosen type within the DBpedia, you should query its CBD.

### 2. Represent the CBD as vector

The CBD comprises triples with the resource  $r_i$  as subject. Removing the common subject from these triples will transform them into a sequence of (predicate, object) pairs. The objects can have two different types: either resources or literals (please note that we exclude blank nodes). Hence, a pair can be represented by three vectors: one vector representing the predicate and two vectors representing the object. The first object vector is either the vector for the object resource or the Null vector and the second vector is either the vector for the object literal or the Null vector.

The single vectors are retrieved from embedding spaces. For properties and resources, we use PYKE.<sup>1</sup> For literals, we use a precomputed Word2Vec model.


---

<sup>1</sup> See how to generate the embeddings space for DBpedia using PYKE  
[https://github.com/dice-group/PYKE/blob/master/PYKE\\_DBpedia.ipynb](https://github.com/dice-group/PYKE/blob/master/PYKE_DBpedia.ipynb)

After that, each pair of the CBD is represented using three vectors. These three vectors can be concatenated to have one single vector for each pair. The stream of these vectors should be merged to a single vector using a transformer.<sup>2</sup>

For later usage, one CBD vector for each resource of the given type is needed.

### 3. Represent questions as vectors

The questions are a sequence of words. Hence, they can be represented using a sequence of word2Vec vectors as input to the RNN. Note that an a-priori disambiguation can be used to spot resources in the question (e.g., using DBpedia Spotlight) 

### 4. Train an RNN

The RNN should be trained based on the training data of the QALD challenge. To this end, the QALD questions have to be filtered to have exactly one result resource and the resource should be of the given type. The input to the RNN is the stream of vectors of the given question and the expected output is the CBD vector of the answer resource.

### 5. Evaluation

Use the test data of the QALD challenge (filtered in the same way as above) and ask the RNN for the CBD vector of the answer of the given question. Use the vector to search for the resources with the highest cosine similarity. This resource is the answer of the system for the given question. Evaluate the approach using GERBIL QA.

---

<sup>2</sup> <https://arxiv.org/pdf/1706.03762.pdf>