July
17

Weekly note

Tags	Weekly
Reference	
Date	
Status	Not started

Knowledge graph updates

For what concerns the modeling of ACT2 data to make them compliant with OpenCitations data model (Link to the data model), I have used RDFLib. I have used a random `base_url` to define the entities of the KG, and I have also built a random structure for the resource (is it correct**???** Should I store the graph in some database**???**).

There are some problems with both the model and data.

The first problem came out because I do not know what some classes should contain, such as:

- `BibliographicReference` → At first, I've used it to store some general information about papers (which are generically mapped into elements of the `Expression` class *for now*) in a dictionary structure, such as the author's name, the title, and the abstract.

  Then, when I tried to convert the RDFLib graph into a NetworkX object [**explained later**], the different formats available didn't accept the list or the dictionary data structures for their node representation.

  At the moment, this class of elements is not included neither in the `.ttl` serialization, neither in the NetworkX graph.

- `InTextReferencePointer` → How should I find such pointers? I have not so clear how I am supposed to work with these pointers.

  The only resources that I am working with have at most the abstract and the context of the citation. Moreover, I also have the doubt that such pointers should come from an annotation (not sure about it, I'm asking).

- `SingleLocationPointerList` → the problem with the above class is directly related with this class.

- `Manifestation` → I do not think that this information should be used, in which way could it be relevant if we are working with all PDFs?

- `RoleInTime` → This one is not so clear to me, I do not understand how to use it. For now, I have treated it in the following way:

```
# Authors
uri_author_id = URIRef(base_url + citing_data[paper_id]['author'])
role_id = URIRef(uri_paper_id + '/' + citing_data[paper_id]['author'] + '/roleInTime')

citations_graph.add((uri_paper_id, is_document_context_for, role_id))
citations_graph.add((role_id, RDF.type, ROLE_IN_TIME))
citations_graph.add((role_id, with_role, AUTHOR))
citations_graph.add((role_id, is_role_held_by, uri_author_id))
```

This problem is directly connected with another thing that I do not understood: the fact that the `Role` class has members `Agent`, `Editor` and `Publisher` means that I can use it has shown above? Can I add a triple of the following type: `citations_graph.add((role_id, with_role, AUTHOR))`?

Going back to the `RoleInTime` class, is it correct to build a specific ID to define an instance of a role for an author and to give it the type of `RoleInTime`?

- `Identifier` and `IdentifierScheme` → How am I supposed to work with them? Do I have to create many different *Identifier entity* connected with all the different papers? If so, do I have to define a URI for these entities? For now, I did in the following way:

  ```
  # Identifier
  citations_graph.add((uri_paper_id, has_identifier, uri_core_id))
  citations_graph.add((uri_core_id, has_literal_value, Literal(citing_data[paper_id]['coreID'])))
  citations_graph.add((uri_core_id, has_identifier_scheme, URL)) # Used URL because I don't know what "CORE_ID" is...
  ```

  Moreover, what is the `CoreID` of a paper? And to which *identifier scheme* does it belong? This question is related to the data that I have from ACT2, where **citing** papers are marked with this `coreID` and **cited** papers are instead referenced with a `DOI`.

- `Annotation` → I have no idea about the way in which I am supposed to use this class.

Once seen this, there is another problem connected with classes: I do not know how how to work with the `Citation` class for what concerns the `context` element. There is no direct link that connects the citation with its context, neither any part of the data model that gives me the possibility to specify it as a literal or a `TextChunk` element.

The way in which I dealt with this is the following:

```
citations_graph.add((uri_citation_id, has_content, Literal(citations_data[citation_id]['context'])))
```

I gave a new property to citation entities which connects them with their context. Probably it is wrong but I didn't know how to deal with this…

Another solution could have been to provide such context as an element of the `DiscourseElement` class, but it doesn't seem the best solution, in particular because of the following:

1. I don't have any way to specify the citation context content as a `Literal` inside the graph;
2. Even if it is possible, I should find a way to map it into the elements of the `LocationPointerList` class, which would be connected with the element of the class `InTextReferencePointer`, that allows to express a literal value of its content.
3. Specifying the context in this way would produce different relations between the contexts contained inside a paper (which has more than a single citation), but in this way we would have a *missing link* between the exact citation connected with this context. This would in turn produce embeddings which do not consider the specific contexts of citations as single elements directly linked to them, but only has parts of the paper, deleting the (probably) most important information to the intent classification aim.

Mapping of `hasCitationCharacterisation`

For what concerns the idea to store the intent labels thanks to the `hasCitationCharacterisation` property, I tried to produce a sort of mapping of the different citations functions of the dataset with the object properties in CITO. The mapping is as follows:

```
# POSSIBLE MAPPING -> Here, the mapping is considered between the labels proposed in the ACT-dataset and the ObjectProperties defined
"""
0 - BACKGROUND ->   IRI: http://purl.org/spar/cito/obtainsBackgroundFrom
                    The citing entity obtains background information from the cited entity.

1 - COMPARES_CONTRASTS ->   IRI: http://purl.org/spar/cito/discusses
                            The citing entity discusses statements, ideas or conclusions presented in the cited entity.

2 - EXTENSION ->    IRI: http://purl.org/spar/cito/extends
                    The citing entity extends facts, ideas or understandings presented in the cited entity.

3 - FUTURE ->   IRI: http://purl.org/spar/cito/citesAsPotentialSolution
                The citing entity cites the cited entity as providing or containing a possible solution to the issues being discussed.

4 - MOTIVATION ->   IRI: http://purl.org/spar/cito/citesAsSourceDocument
                    The citing entity cites the cited entity as being the entity from which the citing entity is derived, or about whi
```

```
5 - USES ->     IRI: http://purl.org/spar/cito/usesMethodIn
                The citing entity describes work that uses a method detailed in the cited entity.
"""
```

But i still have some doubts for what concerns the `discusses` property used for the general class `compare/contrast`; and for the `citesAsPotentialSolution` and `citesAsSourceDocument`, properties which do not really expresses the semantic of the labels to which they have been assigned.

Graph problems…

Using RDFLib I am not allowed to save the graph in a file format compliant with its visualization in tools such as **Gephi** or **Cytoscape**.

I have tried to convert the RDFLib graph into a `.gml` or into all the other formats, but each one has some problems. For example, one of them do not accepts lists, another one is not able to deal with URIRef objects, and so on…

I cam out with a temporary solution for the analysis of the network, which is to re-build the network also inside NetworkX, giving edge labels as URIRef. In this way I have been able to visualize the graph, but with some problems connected to the fact that I am not able to properly have a grasp of data because the network is extremely big (as it considers also all the `RDF.type` properties, and all the entities defined such as `Agent`.

How am I supposed to deal with the above problems about graph?

> 💡 Do any of you know a plugin for either **Gephi** or **Cytoscape** that allows to explore and upload RDF graphs?

SCAR-DL (GRASPOS-CIC) Updates

For what concerns the SCAR-DL model, I have looked into it and tried to make it work but some particular problems came out.

First of all, the right **Tensorflow** version seems to be the **.15** and not the .14 as specified in the requirements.

Then, I got stuck because of two main factors:

1. The `Cython` `.so` file is built for intel processors, thus it won't work in M1/M2 architectures.

   ```
   NotFoundError: dlopen(/Users/lorenzo/miniconda3/lib/python3.10/site-packages/tf_sentencepiece/_sentencepiece_processor_ops.so,
    0x0006): tried: '/Users/lorenzo/miniconda3/lib/python3.10/site-packages/tf_sentencepiece/_sentencepiece_processor_ops.so' (not
   a mach-o file), '/System/Volumes/Preboot/Cryptexes/OS/Users/lorenzo/miniconda3/lib/python3.10/site-packages/tf_sentencepiece/_s
   entencepiece_processor_ops.so' (no such file), '/Users/lorenzo/miniconda3/lib/python3.10/site-packages/tf_sentencepiece/_senten
   cepiece_processor_ops.so' (not a mach-o file)
   ```

2. Working on Intel processors, the issue above is solved, but another exception occurs. In particular, the file `cache/USE_MLQA.train.anchorsent.embedding_cache.pkl`, which is not present by default (I think), is not built even though it is called by the `with open('...', 'wb')` that is supposed to build the file if it doesn't exist.

   ```
   FileNotFound: [Errno 2] No such file or directory: 'cache/USE_MLQA.train.anchorsent.embedding_cache.pkl'
   ```

   This error directly derives from the following function:

   ```
   # Call ->
   n_classes = encode_dataset({'train':trainset, 'test':testset})

   # Function ->
   def encode_dataset(dataset):
     # Embed anchor sentences into vectors
     for key,value in dataset.items():
       df = value['x']
       if USE_PATTERN_EMBEDDING:
         df['main_predicate'] = df['anchorsent']
       # Embed anchor sentences
       #df['anchorsent'] = list(df['anchorsent'])
       cache_file = f'cache/{TF_MODEL}.{key}.anchorsent.embedding_cache.pkl'
       if os.path.isfile(cache_file):
         with open(cache_file, 'rb') as f:
           embedded_sentences_dict = pickle.load(f)
           embedded_sentences = [embedded_sentences_dict[s] for s in df['anchorsent']]
   ```

```
    else:
      MODEL_MANAGER = PredicateExtractor(MODEL_OPTIONS)
      embedded_sentences = MODEL_MANAGER.embed(df['anchorsent'])
      with open(cache_file, 'wb') as f:
        pickle.dump(dict(zip(df['anchorsent'],embedded_sentences)), f)
    df['anchorsent_embedding'] = embedded_sentences
    # Embed extra info
    if USE_PATTERN_EMBEDDING:
      cache_file = f'cache/{TF_MODEL}.{key}.extra.embedding_cache.pkl'
      if os.path.isfile(cache_file):
        with open(cache_file, 'rb') as f:
          embedded_extra_dict = pickle.load(f)
          embedded_extra = [embedded_extra_dict[s] for s in df['main_predicate']]
      else:
        MODEL_MANAGER = PredicateExtractor(MODEL_OPTIONS)
        extra_list = []
        for text in df['main_predicate']:
          extra = list(Counter(pattern['predicate'] for pattern in MODEL_MANAGER.get_pattern_list(text)).keys())
          extra_list.append(extra[0] if len(extra)>0 else '')
        embedded_extra = MODEL_MANAGER.embed(extra_list)
        with open(cache_file, 'wb') as f:
          pickle.dump(dict(zip(df['main_predicate'],embedded_extra)), f)
      df['main_predicate'] = embedded_extra

  # Encode labels
  label_encoder_target = LabelEncoder()
  label_encoder_target.fit([e for set in dataset.values() for e in set['y']])
  print('Label classes:', list(label_encoder_target.classes_))
  for set in dataset.values():
    set['y'] = label_encoder_target.transform(set['y'])

  # Encode sectypes
  all_sectypes = [e for set in dataset.values() for e in set['x']['sectype']]
  label_encoder_sectype = LabelEncoder()
  all_sectypes = label_encoder_sectype.fit_transform(all_sectypes)
  onehot_encoder_sectype = OneHotEncoder()
  onehot_encoder_sectype.fit(all_sectypes.reshape(-1, 1))
  print('SCAR classes:', list(label_encoder_sectype.classes_))
  for set in dataset.values():
    labeled_sectypes = label_encoder_sectype.transform(set['x']['sectype'])
    set['x']['sectype'] = onehot_encoder_sectype.transform(labeled_sectypes.reshape(-1, 1)).toarray()[:,1:]

  # Input features to numpy array
  for set in dataset.values():
    numpyfy_dataset(set)
  # Return number of target classes
  return len(label_encoder_target.classes_)
```

I have no idea on how to solve such issue…probably professor Di Iorio or Sovrano know something more about it. Try to look at this part, it could be somehow related:

```
# The snippet can be found in the "Model Manager" file
MODULE_URL = {
  'USE_Transformer': {
    'local':'/Users/toor/Desktop/NLP/Tutorials/Sentence Embedding/Universal Sentence Encoder/slow',
    'remote': 'https://tfhub.dev/google/universal-sentence-encoder-large/3',
  },
  'USE_DAN': {
    'local':'/Users/toor/Desktop/NLP/Tutorials/Sentence Embedding/Universal Sentence Encoder/fast',
    'remote': 'https://tfhub.dev/google/universal-sentence-encoder/2',
  },
  'USE_MLQA': {
    'local':'/Users/toor/Desktop/NLP/Tutorials/Sentence Embedding/Universal Sentence Encoder/multilingual-qa',
    'remote': 'https://tfhub.dev/google/universal-sentence-encoder-multilingual-qa/1',
  },
}
```

3. Another issue arises from the fact that the library `tf_metrics` is only compatible with older versions of TensorFlow (`<= 1.15`). While this problem can potentially be solved by using `tf.keras.metrics`, I have yet to do a trial run and see if this is a viable solution.
4. Finally, thinking about the dataset, is probably most convenient to use the ACT dataset, instead than the others. Also the idea of combining more dataset seems to require a bit of mapping between citation intents, which is not so easy to implement and to define.

Seen all of this, in my opinion it would be better to start again and define maybe a similar model, or a different one. Trying to rebuild this same exact software for a more recent Tensorflow version could came out to be way more difficult than expected, in

particular because I am not the author, thus it's not so easy for me to understand all the preprocessing functions and the parts connected with Cython.

Furthermore, we could directly start working on *Colab* or *Kaggle*, making way easier to test the model. The idea by itself seems to be really good, the only problem could be to adapt the ACT dataset to this software, which is built on different data.

Dataset

I need the original dataset, or I won't be able to adapt anything to it. The toy dataset ACT2 is not complete, and it has been probably built with some missing data because of the competition, but I'm pretty confident that in ACT data are not missing. Also some DOIs are not recognized as URI, and I'm not so sure about their validity as identifier, are the following compliant forms?

```
http://example.org/CIC_entity/10.1644/1545-1542(2001)082<0960:atotgs>2.0.co;2 does not look like a valid URI, trying to serialize t
his will break. -> 10.1644/1545-1542
http://example.org/CIC_entity/10.1002/1097-4571(2000)9999:9999<::aid-asi1564>3.0.co;2-l does not look like a valid URI, trying to s
erialize this will break. -> 10.1002/1097-4571
http://example.org/CIC_entity/10.1002/(sici)1097-0258(19970415)16:7<791::aid-sim500>3.0.co;2-# does not look like a valid URI, tryi
ng to serialize this will break. -> 10.1002/1097-0258
http://example.org/CIC_entity/10.1175/1520-0477(1996)077<0437:tnyrp>2.0.co;2 does not look like a valid URI, trying to serialize th
is will break. -> 10.1175/1520-0477
http://example.org/CIC_entity/10.1002/(sici)1096-9861(19971020)387:2<167::aid-cne1>3.0.co;2-z does not look like a valid URI, tryin
g to serialize this will break. -> 10.1002/1096-9861
http://example.org/CIC_entity/10.1175/1520-0469(2002)059<0140:eboaai>2.0.co;2 does not look like a valid URI, trying to serialize t
his will break. -> 10.1175/1520-0469
http://example.org/CIC_entity/10.1175/1520-0469(2002)059<0125:eboaai>2.0.co;2 does not look like a valid URI, trying to serialize t
his will break. -> 10.1175/1520-0469
http://example.org/CIC_entity/10.1002/(sici)1099-1166(199906)14:6<481::aid-gps959>3.0.co;2-5 does not look like a valid URI, trying
to serialize this will break. -> 10.1002/1099-1166
http://example.org/CIC_entity/10.1002/1097-0258(20001130)19:22<3127::aid-sim784>3.0.co;2-m does not look like a valid URI, trying t
o serialize this will break. -> 10.1002/1097-0258
http://example.org/CIC_entity/10.1002/(sici)1520-6394(1999)10:3<89::aid-da1>3.0.co;2-5 does not look like a valid URI, trying to se
rialize this will break. -> 10.1002/1520-6394
http://example.org/CIC_entity/10.1002/1097-0193(200101)12:1<1::aid-hbm10>3.0.co;2-v does not look like a valid URI, trying to seria
lize this will break. -> 10.1002/1097-0193
http://example.org/CIC_entity/10.1002/(sici)1099-0984(199612)10:5<303::aid-per262>3.3.co;2-u does not look like a valid URI, trying
to serialize this will break. -> 10.1002/1099-0984
http://example.org/CIC_entity/10.1002/1097-0142(20010415)91:8+<1636::aid-cncr1176>3.0.co;2-d does not look like a valid URI, trying
to serialize this will break. -> 10.1002/1097-0142
http://example.org/CIC_entity/10.1002/1520-6750(199206)39:4<447::aid-nav3220390403>3.0.co;2-o does not look like a valid URI, tryin
g to serialize this will break. -> 10.1002/1520-6750
http://example.org/CIC_entity/10.1002/(sici)1097-0266(200003)21:3<203::aid-smj102>3.0.co;2-k does not look like a valid URI, trying
to serialize this will break. -> 10.1002/1097-0266
http://example.org/CIC_entity/10.1002/(sici)1096-9837(199603)21:3<217::aid-esp611>3.0.co;2-u does not look like a valid URI, trying
to serialize this will break. -> 10.1002/1096-9837
```

Readings and material

I have not started to read the material about KGE, I will do it when I will have a better idea about data preprocessing, in this way I will also be able to make some experiments.