Reproduction of Experimental Results: ViQuAE, a Dataset for Knowledge-Based Visual Question Answering about Named Entities

Introduction

The following procedure is used to reproduce the experiment for the ViQuAE paper(https://github.com/PaulLerner/ViQuAE.git). It is based on the original script from the ViQuAE repository, but has been modified in some places to make it work with the current version of the code. This modifications will be mentioned in the comments.

Authors

- Saban Akay*
- Fabian Ombui*
- Kamogelo Dorcus Mookantsa*
- Elton Tinashe Mpofu*

* All authors are affiliated with Vienna University of Technology, Vienna, Austria.

Experiment Date

```
January 2024
```

Experiment Setup

Code Retrieval

Clone the repository:

```
git clone https://github.com/PaulLerner/ViQuAE.git
```

Then implement the experiment within the cloned repository.

Install Requirements

The following packages are needed to run the code by the ViQuAE repository:

```
datasets>=2.4.0
docopt>=0.6.2
facenet-pytorch>=2.5.2
SPARQLWrapper>=1.8.5
tabulate>=0.8.9
tqdm>=4.49.0
seaborn>=0.11
spacy>=2.2.4
elasticsearch>=7.7.1
transformers>=4.22.2
optuna>=2.9.1
scikit-image
opencv-python>=4.5.3
faiss-gpu>=1.7.1
ranx>=0.3.2
pytorch-lightning[extra]>=1.7.7
```

In [ ]:
```
!python -m pip install -r requirements.txt
```

```
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Requirement already satisfied: datasets>=2.4.0 in ~\python310\lib\site-packages (fro
m -r requirements.txt (line 1)) (2.16.1)
...
WARNING: Ignoring invalid distribution - (~\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (~\python310\lib\site-packages)
ERROR: Could not find a version that satisfies the requirement faiss-gpu>=1.7.1 (fro
m versions: none)
ERROR: No matching distribution found for faiss-gpu>=1.7.1
```

Create a new requirements_dev.txt file without "faiss-gpu>=1.7.1" and install the requirements_dev.txt file with the following command:

```
grep -v "faiss-gpu>=1.7.1" requirements.txt > requirements_dev.txt
pip install -r requirements_dev.txt
```

We tried to install the dependencies with the help of the documentation, but we couldn't install the faiss-gpu dependency. We tried to install faiss-cpu, but it also didn't work. We checked the faiss documentation, read the issues on github, but we couldn't find a solution for this dependency. Therefore, we decided to move on the experiment without it.

Please note that we tried to reproduce the experiment on a Windows machine with an 32GB processor without GPU usage. In the paper **ViQuAE, a Dataset for Knowledge-Based Visual Question Answering about Named Entities. Paul Lerner, Olivier Ferret, Camille Guinaudeau, Hervé Le Borgne, Romaric Besançon, Jose G. Moreno, and Jesús Lovón Melgarejo (SIGIR 2022)**, it has been specified about the usage of powerful setup as **All experiments were carried out with NVIDIA V100 GPUs with 32GB of RAM**. The *s* after the **GPU** was then cleared in the following section as **We train DPR using 4 V100 GPUs of 32GB**.

Data Retrieval and Metadata Collection

Data retrieval and metadata collection are done by the following file: README.rst from the ViQuAE repository. The following commands are used to download the data and metadata:

In [ ]:
```
!git clone https://huggingface.co/datasets/PaulLerner/viquae_all_images
```

```
fatal: destination path 'viquae_all_images' already exists and is not an empty direc
tory.
```

In [ ]:
```
!git clone https://huggingface.co/datasets/PaulLerner/viquae_images
```

```
fatal: destination path 'viquae_images' already exists and is not an empty director
y.
```

In [ ]:
```
!git clone https://huggingface.co/datasets/PaulLerner/viquae_dataset
```

```
Cloning into 'viquae_dataset'...
Updating files:  80% (4/5)
Updating files: 100% (5/5)
Updating files: 100% (5/5), done.
Filtering content:  66% (2/3)
Filtering content: 100% (3/3)
Filtering content: 100% (3/3), 8.26 MiB | 1.82 MiB/s, done.
```

In [ ]:
```
!git clone https://huggingface.co/datasets/PaulLerner/viquae_wikipedia
```

```
fatal: destination path 'viquae_wikipedia' already exists and is not an empty direct
ory.
```

In [ ]:
```
from datasets import load_dataset
dataset = load_dataset('PaulLerner/viquae_dataset')
dataset
```

```
~\Python310\lib\site-packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Ple
ase update jupyter and ipywidgets. See https://ipywidgets.readthedocs.io/en/stable/u
ser_install.html
  from .autonotebook import tqdm as notebook_tqdm
~\Python310\lib\site-packages\huggingface_hub\repocard.py:105: UserWarning: Repo car
d metadata block was not found. Setting CardData to empty.
  warnings.warn("Repo card metadata block was not found. Setting CardData to empt
y.")
```

Out[ ]:
```
DatasetDict({
    train: Dataset({
        features: ['image', 'input', 'kilt_id', 'id', 'meta', 'original_question',
'output', 'url', 'wikidata_id'],
        num_rows: 1190
    })
    validation: Dataset({
        features: ['image', 'input', 'kilt_id', 'id', 'meta', 'original_question',
'output', 'url', 'wikidata_id'],
        num_rows: 1250
    })
    test: Dataset({
        features: ['image', 'input', 'kilt_id', 'id', 'meta', 'original_question',
'output', 'url', 'wikidata_id'],
        num_rows: 1257
    })
})
```

In [ ]:
```
# save the dataset to a file
dataset.save_to_disk('data/viquae_dataset')
```

```
Saving the dataset (1/1 shards): 100%|██████████| 1190/1190 [00:00<00:00, 93501.84 e
xamples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1250/1250 [00:00<00:00, 126762.09
examples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1257/1257 [00:00<00:00, 1936188.07
examples/s]
```

In [ ]:
```
item = dataset['test'][0]
```

In [ ]:
```
type(item)
```

Out[ ]:
```
dict
```

In [ ]:
```
item['url']
```

Out[ ]:
```
'http://upload.wikimedia.org/wikipedia/commons/thumb/a/ae/Jackie_Wilson.png/512px-
Jackie_Wilson.png'
```

In [ ]:
```
item['image']
```

Out[ ]:
```
'512px-Jackie_Wilson.png'
```

The code was looking for a *Commons* directory in the *data* directory which was not available in the repository. We implemented a new line of code to specify the directory of the images. By commenting the original code, the new line of code is as follows:

```
IMAGE_PATH = Path(os.environ.get("VIQUAE_IMAGES_PATH", COMMONS_PATH))
#IMAGE_PATH =
(Path(ROOT_PATH).parent.parent/"viquae_images/images").resolve()
```

Following code is used to set the directory of the images:

In [ ]:
```
import os
os.environ['VIQUAE_IMAGES_PATH']='viquae_images/images'
```

After the implementation of the new line of code, the following command is used to check the validity of the images:

In [ ]:
```
from meerqat.data.loading import load_image
load_image(item['image'])
```

Out[ ]:

In [ ]:
```
from meerqat.data.loading import load_image_batch
image = load_image_batch([item['image']])[0]
type(image), image.size
```

Out[ ]:
```
(PIL.Image.Image, (512, 526))
```

In [ ]:
```
item['input']
```

Out[ ]:
```
"this singer's re-issued song became the UK Christmas number one after helping to
advertise what brand?"
```

In [ ]:
```
item['output']['original_answer']
```

Out[ ]:
```
"Levi's"
```

So far, the images are downloaded and the metadata is collected. The next step is to create

the wikipedia dataset. The following command is used to create the wikipedia dataset:

In [ ]:
```
data_files = dict(
    humans_with_faces='humans_with_faces.jsonl.gz',
    humans_without_faces='humans_without_faces.jsonl.gz',
    non_humans='non_humans.jsonl.gz'
)
kb = load_dataset('PaulLerner/viquae_wikipedia', data_files=data_files)
kb
```

```
~\Python310\lib\site-packages\huggingface_hub\repocard.py:105: UserWarning: Repo car
d metadata block was not found. Setting CardData to empty.
  warnings.warn("Repo card metadata block was not found. Setting CardData to empt
y.")
```

Out[ ]:
```
DatasetDict({
    humans_with_faces: Dataset({
        features: ['anchors', 'categories', 'image', 'kilt_id', 'text', 'url', 'wi
kidata_info', 'wikipedia_id', 'wikipedia_title'],
        num_rows: 506237
    })
    humans_without_faces: Dataset({
        features: ['anchors', 'categories', 'image', 'kilt_id', 'text', 'url', 'wi
kidata_info', 'wikipedia_id', 'wikipedia_title'],
        num_rows: 35736
    })
    non_humans: Dataset({
        features: ['anchors', 'categories', 'image', 'kilt_id', 'text', 'url', 'wi
kidata_info', 'wikipedia_id', 'wikipedia_title'],
        num_rows: 953379
    })
})
```

In [ ]:
```
kb.save_to_disk('data/viquae_wikipedia')
```

```
Saving the dataset (8/8 shards): 100%|██████████| 506237/506237 [00:04<00:00, 10945
0.52 examples/s]
Saving the dataset (1/1 shards): 100%|██████████| 35736/35736 [00:00<00:00, 137622.3
3 examples/s]
Saving the dataset (13/13 shards): 100%|██████████| 953379/953379 [01:40<00:00, 952
5.74 examples/s]
```

In [ ]:
```
item = kb['humans_with_faces'][0]
item['wikidata_info']['wikidata_id'], item['wikidata_info']['wikipedia_title']
```

Out[ ]:
```
('Q313590', 'Alain Connes')
```

In [ ]:
```
item['image']
```

Out[ ]:
```
'512px-Alain_Connes.jpg'
```

In [ ]:
```
# the text is stored in a list of string, one per paragraph
type(item['text']['paragraph']), len(item['text']['paragraph'])
```

Out[ ]:
```
(list, 25)
```

In [ ]:
```
# the text is stored in a list of string, one per paragraph
item['text']['paragraph'][1]
```

Out[ ]:
```
'Alain Connes (; born 1 April 1947) is a French mathematician, currently Professor
at the Collège de France, IHÉS, Ohio State University and Vanderbilt University. H
e was an Invited Professor at the Conservatoire national des arts et métiers (200
0).\n'
```

Next step is to combine The ViQuAE Knowledge Base. The following command is used to combine the datasets:

In [ ]:
```
#to concatenate these three datasets to get a single dataset (e.g. to split the art
from datasets import concatenate_datasets
kb['humans_with_faces'] = kb['humans_with_faces'].map(lambda item: {'is_human': Tru
kb['humans_without_faces'] = kb['humans_without_faces'].map(lambda item: {'is_human
kb['non_humans'] = kb['non_humans'].map(lambda item: {'is_human': False})
kb_recat = concatenate_datasets([kb['non_humans'], kb['humans_with_faces'], kb['hum
kb_recat.save_to_disk('data/viquae_wikipedia_recat')
```

```
Map: 100%|████████████████████████████████████████████████████████████████
████████████████████████████████| 506237/506237 [01:09<00:00, 7303.40 examples/s]
Map: 100%|████████████████████████████████████████████████████████████████
████████████████████████████████| 35736/35736 [00:05<00:00, 7114.51 examples/s]
Map: 100%|████████████████████████████████████████████████████████████████
████████████████████████████████| 953379/953379 [02:27<00:00, 6467.61 examples/s]
Saving the dataset (22/22 shards): 100%|████████████████████████████████████
████████████████████████████████| 1495352/1495352 [03:17<00:00, 7587.34 examples/s]
```

Experiment Reproduction

From here onwards, the experiment is reproduced with the help of the EXPERIMENT.rst file from the ViQuAE repository. The following commands are used to reproduce the experiment:

Preprocessing Passages

Splitting articles in passages

So far, all the data is downloaded and combined. The next step is to create the passages.

In [ ]:
```
!python -m meerqat.data.loading passages data/viquae_wikipedia_recat data/viquae_pa
```

```
Dataset({
    features: ['passage', 'index'],
    num_rows: 11895924
})
```

```
~\edds\meerqat\data\loading.py:458: FutureWarning: set_caching_enabled is deprecated
and will be removed in the next major version of datasets. Use datasets.enable_cachi
ng() or datasets.disable_caching() instead. This function will be removed in a futur
e version of datasets.
  set_caching_enabled(not args['--disable_caching'])

Map:   0%|          | 0/1495352 [00:00<?, ? examples/s]
Map:   0%|          | 2/1495352 [00:00<25:53:26, 16.04 examples/s]
...
Saving the dataset (22/22 shards): 100%|##########| 1495352/1495352 [03:53<00:00, 63
94.11 examples/s]
```

With the setup applied so far, even though it took a long time, we were able to split the passages. There was no information about the validity of the passages in the documentation, nor any indication of how to check it. Therefore, we moved on to the next step since each attempt took a long time.

Following commands are used to *extract some columns from the dataset to allow quick (and string) indexing*:

In [ ]:
```
!python -m meerqat.data.loading map data/viquae_wikipedia_recat wikipedia_title tit
```

```
~\edds\meerqat\data\loading.py:458: FutureWarning: set_caching_enabled is deprecated
and will be removed in the next major version of datasets. Use datasets.enable_cachi
ng() or datasets.disable_caching() instead. This function will be removed in a futur
e version of datasets.
  set_caching_enabled(not args['--disable_caching'])

Map:   0%|          | 0/1495352 [00:00<?, ? examples/s]
...
Map: 100%|##########| 1495352/1495352 [00:36<00:00, 41531.46 examples/s]
```

In [ ]:
```
!python -m meerqat.data.loading map data/viquae_wikipedia_recat passage_index artic
```

```
~\edds\meerqat\data\loading.py:458: FutureWarning: set_caching_enabled is deprecated
and will be removed in the next major version of datasets. Use datasets.enable_cachi
ng() or datasets.disable_caching() instead. This function will be removed in a futur
e version of datasets.
  set_caching_enabled(not args['--disable_caching'])

Map:   0%|          | 0/1495352 [00:00<?, ? examples/s]
Map:   0%|          | 3232/1495352 [00:00<00:47, 31160.19 examples/s]
...
Map: 100%|##########| 1495352/1495352 [00:45<00:00, 33048.55 examples/s]
```

Both codes were implemented without any problems.

Find relevant passages in the linked wikipedia articles

In [ ]:
```
!python -m meerqat.ir.metrics relevant data/viquae_dataset data/viquae_passages dat
```

```
Map:   0%|          | 0/1190 [00:00<?, ? examples/s]
Map:   0%|          | 1/1190 [00:00<02:22,  8.34 examples/s]
...
Map: 100%|█████████| 1256/1257 [01:11<00:00, 19.29 examples/s]
Map: 100%|█████████| 1257/1257 [01:11<00:00, 17.47 examples/s]
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\ir\metrics.py", line 406, in <module>
    find_relevant_dataset(
  File "~\edds\meerqat\ir\metrics.py", line 195, in find_relevant_dataset
    dataset.save_to_disk(dataset_path)
  File "~\Python310\lib\site-packages\datasets\dataset_dict.py", line 1289, in save_
to_disk
    dataset.save_to_disk(
  File "~\Python310\lib\site-packages\datasets\arrow_dataset.py", line 1471, in save
_to_disk
    raise PermissionError(
PermissionError: Tried to overwrite ~\edds\data\viquae_dataset\train but a dataset c
an't overwrite itself.
```

The method implemented for this step is requiring an overwriting of the original dataset. There were no explanations about this issue in the documentation. We tried to overwrite it with different implementations. But did not make sense. Therefore, we decided to create a new dataset with the following changes applied to the original *meerqat/ir/metrics.py* file:

```
195     dataset.save_to_disk(dataset_path)   # Original code
195     dataset.save_to_disk(str(dataset_path) + "/new")   # New
code(WindowsPath required a string)
```

In [ ]:
```
!python -m meerqat.ir.metrics relevant data/viquae_dataset data/viquae_passages dat
```

```
Saving the dataset (0/1 shards):   0%|          | 0/1190 [00:00<?, ? examples/s]
Saving the dataset (0/1 shards):  84%|████████▍  | 1000/1190 [00:01<00:00, 541.51 ex
amples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1190/1190 [00:01<00:00, 541.51 exa
mples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1190/1190 [00:01<00:00, 617.07 exa
mples/s]

Saving the dataset (0/1 shards):   0%|          | 0/1250 [00:00<?, ? examples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1250/1250 [00:00<00:00, 86105.54 e
xamples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1250/1250 [00:00<00:00, 86105.54 e
xamples/s]

Saving the dataset (0/1 shards):   0%|          | 0/1257 [00:00<?, ? examples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1257/1257 [00:00<00:00, 86338.17 e
xamples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1257/1257 [00:00<00:00, 86338.17 e
xamples/s]
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\ir\metrics.py", line 406, in <module>
    find_relevant_dataset(
  File "~\edds\meerqat\ir\metrics.py", line 199, in find_relevant_dataset
    qrel = ranx.Qrels({q_id: qrels[q_id] for q_id in subset['id']})
  File "~\edds\meerqat\ir\metrics.py", line 199, in <dictcomp>
    qrel = ranx.Qrels({q_id: qrels[q_id] for q_id in subset['id']})
KeyError: 'a79b04bb9ea4c1e17edf31d544fd17bd'
```

WE had no idea about what is there is an attempt to access a key ('a79b04bb9ea4c1e17edf31d544fd17bd') in a dictionary (qrels) that doesn't exist. The setup and experimental steps has been robustly followed so far. We tried to find the reason for this error, but we couldn't find it. Therefore, we decided move on to the next step.

Find relevant passages in the IR results

In this step, the following code is designed to improve the accuracy of information retrieval by filtering out irrelevant passages obtained during the Information Retrieval (IR) step and gives priority to passages that align with the original answer.

The path of the *bm25* file is changed to the following:

```
#run = Run.from_file('/path/to/bm25/or/multimodal_ir_train.trec')
run = Run.from_file('experiments/ir/viquae/bm25/BM25.trec')
```

In [ ]:
```
from datasets import load_from_disk, set_caching_enabled
from meerqat.ir.metrics import find_relevant
from ranx import Run

set_caching_enabled(False)
```

```python
kb = load_from_disk('data/viquae_passages/')
dataset = load_from_disk('data/viquae_dataset/train')
# to reproduce the results of the papers:
# - use DPR+Image as IR to train the reader or fine-tune ECA/ILF
# - use BM25 as IR to train DPR (then save in 'BM25_provenance_indices'/'BM25_irrel
run = Run.from_file('experiments/ir/viquae/bm25/BM25.trec')

def keep_relevant_search_wrt_original_in_priority(item, kb):
    indices = list(map(int, run[item['id']]))
    relevant_indices, _ = find_relevant(indices, item['output']['original_answer'],
    if relevant_indices:
        item['BM25_provenance_indices'] = relevant_indices
    else:
        item['BM25_provenance_indices'] = item['original_answer_provenance_indices'
    item['BM25_irrelevant_indices'] = list(set(indices) - set(relevant_indices))
    return item

dataset = dataset.map(keep_relevant_search_wrt_original_in_priority, fn_kwargs=dict
dataset.save_to_disk('data/viquae_dataset/train')
```

```
Map:   0%|          | 0/1190 [00:00<?, ? examples/s]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[39], line 23
     20     item['BM25_irrelevant_indices'] = list(set(indices) - set(relevant_indic
es))
     21     return item
---> 23 dataset = dataset.map(keep_relevant_search_wrt_original_in_priority, fn_kwar
gs=dict(kb=kb))
     24 dataset.save_to_disk('data/viquae_dataset/train')

File ~\Python310\lib\site-packages\datasets\arrow_dataset.py:592, in transmit_tasks.
<locals>.wrapper(*args, **kwargs)
    590     self: "Dataset" = kwargs.pop("self")
    591 # apply actual function
--> 592 out: Union["Dataset", "DatasetDict"] = func(self, *args, **kwargs)
    593 datasets: List["Dataset"] = list(out.values()) if isinstance(out, dict) else
[out]
    594 for dataset in datasets:
    595     # Remove task templates if a column mapping of the template is no longer
valid

File ~\Python310\lib\site-packages\datasets\arrow_dataset.py:557, in transmit_forma
t.<locals>.wrapper(*args, **kwargs)
    550 self_format = {
    551     "type": self._format_type,
    552     "format_kwargs": self._format_kwargs,
    553     "columns": self._format_columns,
    554     "output_all_columns": self._output_all_columns,
    555 }
    556 # apply actual function
--> 557 out: Union["Dataset", "DatasetDict"] = func(self, *args, **kwargs)
    558 datasets: List["Dataset"] = list(out.values()) if isinstance(out, dict) else
[out]
    559 # re-apply format to the output

File ~\Python310\lib\site-packages\datasets\arrow_dataset.py:3093, in Dataset.map(se
lf, function, with_indices, with_rank, input_columns, batched, batch_size, drop_last
_batch, remove_columns, keep_in_memory, load_from_cache_file, cache_file_name, write
r_batch_size, features, disable_nullable, fn_kwargs, num_proc, suffix_template, new_
fingerprint, desc)
   3087 if transformed_dataset is None:
   3088     with hf_tqdm(
   3089         unit=" examples",
   3090         total=pbar_total,
   3091         desc=desc or "Map",
   3092     ) as pbar:
-> 3093         for rank, done, content in Dataset._map_single(**dataset_kwargs):
   3094             if done:
   3095                 shards_done += 1

File ~\Python310\lib\site-packages\datasets\arrow_dataset.py:3446, in Dataset._map_s
ingle(shard, function, with_indices, with_rank, input_columns, batched, batch_size,
 drop_last_batch, remove_columns, keep_in_memory, cache_file_name, writer_batch_siz
e, features, disable_nullable, fn_kwargs, new_fingerprint, rank, offset)
   3444 _time = time.time()
   3445 for i, example in shard_iterable:
```

```
-> 3446     example = apply_function_on_filtered_inputs(example, i, offset=offset)
   3447     if update_data:
   3448         if i == 0:

File ~\Python310\lib\site-packages\datasets\arrow_dataset.py:3349, in Dataset._map_s
ingle.<locals>.apply_function_on_filtered_inputs(pa_inputs, indices, check_same_num_
examples, offset)
   3347 if with_rank:
   3348     additional_args += (rank,)
-> 3349 processed_inputs = function(*fn_args, *additional_args, **fn_kwargs)
   3350 if isinstance(processed_inputs, LazyDict):
   3351     processed_inputs = {
   3352         k: v for k, v in processed_inputs.data.items() if k not in processed
_inputs.keys_to_format
   3353     }

Cell In[39], line 14, in keep_relevant_search_wrt_original_in_priority(item, kb)
     13 def keep_relevant_search_wrt_original_in_priority(item, kb):
---> 14     indices = list(map(int, run[item['id']]))
     15     relevant_indices, _ = find_relevant(indices, item['output']['original_an
swer'], [], kb)
     16     if relevant_indices:

File ~\Python310\lib\site-packages\ranx\data_structures\run.py:383, in Run.__getitem
__(self, q_id)
    382 def __getitem__(self, q_id):
--> 383     return dict(self.run[q_id])

File ~\Python310\lib\site-packages\numba\typed\typeddict.py:180, in Dict.__getitem__
(self, key)
    178     raise KeyError(key)
    179 else:
--> 180     return _getitem(self, key)

File ~\Python310\lib\site-packages\numba\typed\dictobject.py:778, in impl()
    776 ix, val = _dict_lookup(d, castedkey, hash(castedkey))
    777 if ix == DKIX.EMPTY:
--> 778     raise KeyError()
    779 elif ix < DKIX.EMPTY:
    780     raise AssertionError("internal dict error during lookup")

KeyError:
```

It gave a *KeyError* with no explanation. We tried to find the reason for this error, but we couldn't find it. Therefore, we decided try *DPR+Image* for the given script. There supposed to be a *multimodal_ir_train.trec* file in the *experiments/ir/viquae* directory, but it was not available. Only *.trec* files are:

- experiments/ir/viquae/bm25/BM25.trec
- experiments/ir/viquae/bm25+arcface+clip+imagenet/fusion.trec
- experiments/ir/viquae/dpr/dpr+arcface+clip+imagenet/fusion.trec
- experiments/ir/viquae/dpr/eca/embedding/Eca_few_shot.trec

Since these files are not described in the documentation, we decided to move on to the next step.

Image Embedding

In [ ]:
```
# embed dataset images with ImageNet-ResNet50
!python -m meerqat.image.embedding data/viquae_dataset experiments/image_embedding/
```

```
Loaded pre-trained model on ImageNet
ImageEncoder(
  (encoder): Sequential(
    (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=F
alse)
    (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats
=True)
    (relu): ReLU(inplace=True)
    (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=F
alse)
    (layer1): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
tats=True)
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), b
ias=False)
        ...
        (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running
_stats=True)
        (relu): ReLU(inplace=True)
      )
    )
  )
  (pool): AdaptiveMaxPool2d(output_size=1)
)
```

```
~\edds\meerqat\image\embedding.py:188: FutureWarning: set_caching_enabled is deprecated and will be removed in the next major version of datasets. Use datasets.enable_caching() or datasets.disable_caching() instead. This function will be removed in a future version of datasets.
  set_caching_enabled(not args['--disable_caching'])
~\Python310\lib\site-packages\torchvision\models\_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn(
~\Python310\lib\site-packages\torchvision\models\_utils.py:223: UserWarning: Arguments other than a weight enum or `None` for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing `weights=ResNet50_Weights.IMAGENET1K_V1`. You can also use `weights=ResNet50_Weights.DEFAULT` to get the most up-to-date weights.
  warnings.warn(msg)

Map:   0%|          | 0/1190 [00:00<?, ? examples/s]~\edds\meerqat\data\loading.py:114: UserWarning: Caught exception '[Errno 2] No such file or directory: '~\\edds\\viquae_images\\images\\512px-John_R._Neill_-_Les_Misérables_-_Cosette_in_front_of_the_doll_shop.jpg'' with image '~\edds\viquae_images\images\512px-John_R._Neill_-_Les_Misérables_-_Cosette_in_front_of_the_doll_shop.jpg'
  warnings.warn(f"Caught exception '{e}' with image '{path}'")
...
Map: 100%|██████████| 1257/1257 [00:57<00:00, 21.73 examples/s]

Saving the dataset (0/1 shards):   0%|          | 0/1190 [00:00<?, ? examples/s]
Saving the dataset (0/1 shards):   0%|          | 0/1190 [00:00<?, ? examples/s]
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\image\embedding.py", line 196, in <module>
    dataset_embed(args['<dataset>'], output_path=args['--output'], **config)
  File "~\edds\meerqat\image\embedding.py", line 183, in dataset_embed
    dataset.save_to_disk(output_path)
  File "~\Python310\lib\site-packages\datasets\dataset_dict.py", line 1289, in save_to_disk
    dataset.save_to_disk(
  File "~\Python310\lib\site-packages\datasets\arrow_dataset.py", line 1535, in save_to_disk
    for job_id, done, content in Dataset._save_to_disk_single(**kwargs):
  File "~\Python310\lib\site-packages\datasets\arrow_dataset.py", line 1559, in _save_to_disk_single
    writer = ArrowWriter(
  File "~\Python310\lib\site-packages\datasets\arrow_writer.py", line 336, in __init__
    self.stream = self._fs.open(fs_token_paths[2][0], "wb")
  File "~\Python310\lib\site-packages\fsspec\spec.py", line 1307, in open
    f = self._open(
  File "~\Python310\lib\site-packages\fsspec\implementations\local.py", line 180, in _open
    return LocalFileOpener(path, mode, fs=self, **kwargs)
  File "~\Python310\lib\site-packages\fsspec\implementations\local.py", line 302, in __init__
    self._open()
```

```
  File "~\Python310\lib\site-packages\fsspec\implementations\local.py", line 307, in
_open
    self.f = open(self.path, mode=self.mode)
OSError: [Errno 22] Invalid argument: '~/edds/data/viquae_dataset/train/data-00000-o
f-00001.arrow'
```

In [ ]:
```
# embed dataset images with CLIP-ResNet50
!python -m meerqat.image.embedding data/viquae_dataset experiments/image_embedding/
```

```
~\edds\meerqat\image\embedding.py:188: FutureWarning: set_caching_enabled is depreca
ted and will be removed in the next major version of datasets. Use datasets.enable_c
aching() or datasets.disable_caching() instead. This function will be removed in a f
uture version of datasets.
  set_caching_enabled(not args['--disable_caching'])
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\image\embedding.py", line 196, in <module>
    dataset_embed(args['<dataset>'], output_path=args['--output'], **config)
  File "~\edds\meerqat\image\embedding.py", line 180, in dataset_embed
    fn_kwargs.update(get_model_and_transform(model_kwargs=model_kwargs, transform_kw
args=transform_kwargs))
  File "~\edds\meerqat\image\embedding.py", line 108, in get_model_and_transform
    clip_model, transform = clip.load(**model_kwargs, device=device)
  File "~\Python310\lib\site-packages\clip\clip.py", line 96, in load
    model.apply(patch_device)
  File "~\Python310\lib\site-packages\torch\nn\modules\module.py", line 897, in appl
y
    module.apply(fn)
  File "~\Python310\lib\site-packages\torch\nn\modules\module.py", line 897, in appl
y
    module.apply(fn)
  File "~\Python310\lib\site-packages\torch\nn\modules\module.py", line 898, in appl
y
    fn(self)
  File "~\Python310\lib\site-packages\clip\clip.py", line 93, in patch_device
    if "value" in node.attributeNames() and str(node["value"]).startswith("cuda"):
TypeError: 'torch._C.Node' object is not subscriptable
```

The following command requires *CLIP* to be installed even though it is not mentioned in the documentation. And it is not just *CLIP* apperanly, but *clip-by-openai* is required. We tried to install it with the following command:

```
!python -m pip install clip-by-openai
```

Then it required *ftfy* to be installed. We tried to install it with the following command:

```
!python -m pip install ftfy
```

In [ ]:
```
# embed KB images with CLIP-ResNet50
!python -m meerqat.image.embedding data/viquae_wikipedia experiments/image_embeddin
```

```
~\edds\meerqat\image\embedding.py:188: FutureWarning: set_caching_enabled is deprecated and will be removed in the next major version of datasets. Use datasets.enable_caching() or datasets.disable_caching() instead. This function will be removed in a future version of datasets.
  set_caching_enabled(not args['--disable_caching'])
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\image\embedding.py", line 196, in <module>
    dataset_embed(args['<dataset>'], output_path=args['--output'], **config)
  File "~\edds\meerqat\image\embedding.py", line 180, in dataset_embed
    fn_kwargs.update(get_model_and_transform(model_kwargs=model_kwargs, transform_kwargs=transform_kwargs))
  File "~\edds\meerqat\image\embedding.py", line 108, in get_model_and_transform
    clip_model, transform = clip.load(**model_kwargs, device=device)
  File "~\Python310\lib\site-packages\clip\clip.py", line 96, in load
    model.apply(patch_device)
  File "~\Python310\lib\site-packages\torch\nn\modules\module.py", line 897, in apply
    module.apply(fn)
  File "~\Python310\lib\site-packages\torch\nn\modules\module.py", line 897, in apply
    module.apply(fn)
  File "~\Python310\lib\site-packages\torch\nn\modules\module.py", line 898, in apply
    fn(self)
  File "~\Python310\lib\site-packages\clip\clip.py", line 93, in patch_device
    if "value" in node.attributeNames() and str(node["value"]).startswith("cuda"):
TypeError: 'torch._C.Node' object is not subscriptable
```

In [ ]:
```
# embed dataset images with CLIP-ViT
!python -m meerqat.image.embedding data/viquae_dataset experiments/image_embedding/
```

```
~\edds\meerqat\image\embedding.py:188: FutureWarning: set_caching_enabled is deprecated and will be removed in the next major version of datasets. Use datasets.enable_caching() or datasets.disable_caching() instead. This function will be removed in a future version of datasets.
  set_caching_enabled(not args['--disable_caching'])
Traceback (most recent call last):
  File "~\Python310\lib\site-packages\transformers\utils\hub.py", line 389, in cached_file
    resolved_file = hf_hub_download(
  File "~\Python310\lib\site-packages\huggingface_hub\utils\_validators.py", line 110, in _inner_fn
    validate_repo_id(arg_value)
  File "~\Python310\lib\site-packages\huggingface_hub\utils\_validators.py", line 158, in validate_repo_id
    raise HFValidationError(
huggingface_hub.utils._validators.HFValidationError: Repo id must be in the form 'repo_name' or 'namespace/repo_name': '/gpfsdswork/dataset/HuggingFace_Models/openai/clip-vit-base-patch32/'. Use `repo_type` argument if needed.

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\image\embedding.py", line 196, in <module>
    dataset_embed(args['<dataset>'], output_path=args['--output'], **config)
  File "~\edds\meerqat\image\embedding.py", line 180, in dataset_embed
    fn_kwargs.update(get_model_and_transform(model_kwargs=model_kwargs, transform_kwargs=transform_kwargs))
  File "~\edds\meerqat\image\embedding.py", line 112, in get_model_and_transform
    model = get_pretrained(**model_kwargs)
  File "~\edds\meerqat\data\loading.py", line 183, in get_pretrained
    model = Class.from_pretrained(pretrained_model_name_or_path, **kwargs)
  File "~\Python310\lib\site-packages\transformers\modeling_utils.py", line 2789, in from_pretrained
    resolved_config_file = cached_file(
  File "~\Python310\lib\site-packages\transformers\utils\hub.py", line 454, in cached_file
    raise EnvironmentError(
OSError: Incorrect path_or_model_id: '/gpfsdswork/dataset/HuggingFace_Models/openai/clip-vit-base-patch32/'. Please provide either the path to a local folder or the repo_id of a model on the Hub.
```

In [ ]:
```
# embed KB images with CLIP-ViT
!python -m meerqat.image.embedding data/viquae_wikipedia experiments/image_embeddin
```

```
~\edds\meerqat\image\embedding.py:188: FutureWarning: set_caching_enabled is deprecated and will be removed in the next major version of datasets. Use datasets.enable_caching() or datasets.disable_caching() instead. This function will be removed in a future version of datasets.
  set_caching_enabled(not args['--disable_caching'])
Traceback (most recent call last):
  File "~\Python310\lib\site-packages\transformers\utils\hub.py", line 389, in cached_file
    resolved_file = hf_hub_download(
  File "~\Python310\lib\site-packages\huggingface_hub\utils\_validators.py", line 110, in _inner_fn
    validate_repo_id(arg_value)
  File "~\Python310\lib\site-packages\huggingface_hub\utils\_validators.py", line 158, in validate_repo_id
    raise HFValidationError(
huggingface_hub.utils._validators.HFValidationError: Repo id must be in the form 'repo_name' or 'namespace/repo_name': '/gpfsdswork/dataset/HuggingFace_Models/openai/clip-vit-base-patch32/'. Use `repo_type` argument if needed.

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\image\embedding.py", line 196, in <module>
    dataset_embed(args['<dataset>'], output_path=args['--output'], **config)
  File "~\edds\meerqat\image\embedding.py", line 180, in dataset_embed
    fn_kwargs.update(get_model_and_transform(model_kwargs=model_kwargs, transform_kwargs=transform_kwargs))
  File "~\edds\meerqat\image\embedding.py", line 112, in get_model_and_transform
    model = get_pretrained(**model_kwargs)
  File "~\edds\meerqat\data\loading.py", line 183, in get_pretrained
    model = Class.from_pretrained(pretrained_model_name_or_path, **kwargs)
  File "~\Python310\lib\site-packages\transformers\modeling_utils.py", line 2789, in from_pretrained
    resolved_config_file = cached_file(
  File "~\Python310\lib\site-packages\transformers\utils\hub.py", line 454, in cached_file
    raise EnvironmentError(
OSError: Incorrect path_or_model_id: '/gpfsdswork/dataset/HuggingFace_Models/openai/clip-vit-base-patch32/'. Please provide either the path to a local folder or the repo_id of a model on the Hub.
```

In [ ]:
```
!python -m meerqat.ir.embedding data/viquae_wikipedia experiments/ir/viquae/clip/co
```

```
~\edds\meerqat\ir\embedding.py:277: FutureWarning: set_caching_enabled is deprecated
and will be removed in the next major version of datasets. Use datasets.enable_cachi
ng() or datasets.disable_caching() instead. This function will be removed in a futur
e version of datasets.
  set_caching_enabled(not args['--disable_caching'])
Traceback (most recent call last):
  File "~\Python310\lib\site-packages\huggingface_hub\utils\_errors.py", line 286, i
n hf_raise_for_status
    response.raise_for_status()
  File "~\Python310\lib\site-packages\requests\models.py", line 1021, in raise_for_s
tatus
    raise HTTPError(http_error_msg, response=self)
requests.exceptions.HTTPError: 401 Client Error: Unauthorized for url: https://huggi
ngface.co/clip-vit-base-patch32/resolve/main/config.json

...

  File "~\Python310\lib\site-packages\transformers\utils\hub.py", line 410, in cache
d_file
    raise EnvironmentError(
OSError: clip-vit-base-patch32 is not a local folder and is not a valid model identi
fier listed on 'https://huggingface.co/models'
If this is a private repository, make sure to pass a token having permission to this
repo either by logging in with `huggingface-cli login` or by passing `token=<your_to
ken>`
```

Face Detection

In this step, no directory adjustments were working. So, we manually generated a *data/Common* filled it with *images* folder from the *viquae_images/images*. Then we returned back to the original path settings:

```
DATA_ROOT_PATH = (Path(ROOT_PATH).parent.parent/"data").resolve()
COMMONS_PATH = DATA_ROOT_PATH / "Commons"
#IMAGE_PATH = Path(os.environ.get("VIQUAE_IMAGES_PATH", COMMONS_PATH))
IMAGE_PATH =
(Path(ROOT_PATH).parent.parent/"viquae_images/images").resolve() # if you
get error regarding this, uncomment above line
```

In [ ]:
```
!python -m meerqat.image.face_detection data/viquae_dataset --disable_caching --bat
```

```
~\edds\meerqat\data\wiki.py:153: UserWarning: ModuleNotFoundError: No module named
'SPARQLWrapper'
  warnings.warn(f"ModuleNotFoundError: {e}")
~\edds\meerqat\image\face_detection.py:160: FutureWarning: set_caching_enabled is de
precated and will be removed in the next major version of datasets. Use datasets.ena
ble_caching() or datasets.disable_caching() instead. This function will be removed i
n a future version of datasets.
  set_caching_enabled(not args['--disable_caching'])

Map:   0%|          | 0/1190 [00:00<?, ? examples/s]~\edds\meerqat\data\loading.py:1
14: UserWarning: Caught exception '[Errno 2] No such file or directory: '~\\edds\\vi
quae_images\\images\\512px-John_R._Neill_-_Les_Misérables_-_Cosette_in_front_of_the_
doll_shop.jpg'' with image '~\edds\viquae_images\images\512px-John_R._Neill_-_Les_Mi
sérables_-_Cosette_in_front_of_the_doll_shop.jpg'
  warnings.warn(f"Caught exception '{e}' with image '{path}'")
~\edds\meerqat\data\loading.py:114: UserWarning: Caught exception '[Errno 2] No such
file or directory: '~\\edds\\viquae_images\\images\\512px-Estatua_de_Pedro_de_Valdiv
ia_en_Cerro_Santa_Lucía.jpg'' with image '~\edds\viquae_images\images\512px-Estatua_
de_Pedro_de_Valdivia_en_Cerro_Santa_Lucía.jpg'
  warnings.warn(f"Caught exception '{e}' with image '{path}'")
~\edds\meerqat\data\loading.py:114: UserWarning: Caught exception '[Errno 22] Invali
d argument: '~\\edds\\viquae_images\\images\\512px-"Là,_Minos_siège,_terrible_et_gro
ndant"_Dessin_de_Gustave_Doré,_gravure_sur_bois_de_Gaston_Monvoisin,_1861_.jpg'' wit
h image '~\edds\viquae_images\images\512px-"Là,_Minos_siège,_terrible_et_grondant"_D
essin_de_Gustave_Doré,_gravure_sur_bois_de_Gaston_Monvoisin,_1861_.jpg'
  warnings.warn(f"Caught exception '{e}' with image '{path}'")
...
~\edds\meerqat\data\loading.py:114: UserWarning: Caught exception '[Errno 2] No such
file or directory: '~\\edds\\viquae_images\\images\\512px-黄河@兰州.jpg'' with image
'~\edds\viquae_images\images\512px-黄河@兰州.jpg'
  warnings.warn(f"Caught exception '{e}' with image '{path}'")
~\edds\meerqat\data\loading.py:114: UserWarning: Caught exception '[Errno 22] Invali
d argument: '~\\edds\\viquae_images\\images\\512px-"Achille_Lauro"_-_Palermo,_1965_
(3).JPG'' with image '~\edds\viquae_images\images\512px-"Achille_Lauro"_-_Palermo,_1
965_(3).JPG'
  warnings.warn(f"Caught exception '{e}' with image '{path}'")

Map:  81%|████████   | 1024/1257 [02:04<00:22, 10.19 examples/s]
Map: 100%|██████████| 1257/1257 [02:04<00:00, 10.96 examples/s]
Map: 100%|██████████| 1257/1257 [02:04<00:00, 10.10 examples/s]

Saving the dataset (0/1 shards):   0%|          | 0/1190 [00:00<?, ? examples/s]
Saving the dataset (0/1 shards):  84%|████████▍  | 1000/1190 [00:09<00:01, 110.19 ex
amples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1190/1190 [00:09<00:00, 110.19 exa
mples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1190/1190 [00:09<00:00, 130.24 exa
mples/s]

Saving the dataset (0/1 shards):   0%|          | 0/1250 [00:00<?, ? examples/s]
Saving the dataset (0/1 shards):  80%|███████   | 1000/1250 [00:00<00:00, 2133.59 ex
amples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1250/1250 [00:00<00:00, 2133.59 ex
amples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1250/1250 [00:00<00:00, 2609.07 ex
amples/s]
```

```
Saving the dataset (0/1 shards):   0%|          | 0/1257 [00:00<?, ? examples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1257/1257 [00:00<00:00, 70076.96 e
xamples/s]
Saving the dataset (1/1 shards): 100%|██████████| 1257/1257 [00:00<00:00, 70076.96 e
xamples/s]
```

In [ ]:
```
!python -m meerqat.image.face_detection data/viquae_wikipedia/humans --disable_cach
```

```
~\edds\meerqat\data\wiki.py:153: UserWarning: ModuleNotFoundError: No module named
'SPARQLWrapper'
  warnings.warn(f"ModuleNotFoundError: {e}")
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\image\face_detection.py", line 158, in <module>
    dataset = load_from_disk(dataset_path)
  File "~\Python310\lib\site-packages\datasets\load.py", line 2630, in load_from_dis
k
    raise FileNotFoundError(f"Directory {dataset_path} not found")
FileNotFoundError: Directory data/viquae_wikipedia/humans not found
```

In this step, a new directory *data/viquae_wikipedia/humans* is found mising. There is no information about this directory in the documentation. We tried to combine data, but this led to missing image issues, and accessing these images wasn't documented. We tried to find the reason for these erros, but we couldn't find it. Therefore, we decided to move on to the next step.

Face Recognition

In [ ]:
```
!git clone https://github.com/PaulLerner/insightface.git
```

```
Cloning into 'insightface'...
Updating files:  18% (135/731)
Updating files:  19% (139/731)
...
Updating files:  99% (724/731)
Updating files: 100% (731/731), done.
```

As specified in the documentation, the following scripts are created to install the *facenet-pytorch* package:

- create a new init file

```
touch insightface/recognition/arcface_torch/__init__.py
```

- add the following line to the init file

```
__import__('pkg_resources').declare_namespace(__name__)
```

- create a new setup file

```
touch insightface/recognition/setup.py
```

- add the following lines to the setup file

```
import os

import pkg_resources
from setuptools import setup, find_packages

with open('arcface_torch/README.md') as f:
    long_description = f.read()

setup(
    name='arcface_torch',
    packages=find_packages(),
    install_requires=[
        str(r)
        for r in pkg_resources.parse_requirements(
            open(os.path.join(os.path.dirname(__file__),
"arcface_torch/requirement.txt"))
        )
    ],

    long_description=long_description,
    long_description_content_type='text/markdown',

    url='https://github.com/deepinsight/insightface',

)
```

An error occured while running the following command:

```
!python -m pip install -e insightface/recognition
```

The error was as follows:

```
        replace 'sklearn' by 'scikit-learn' in your pip requirements
    files
                (requirements.txt, setup.py, setup.cfg, Pipfile, etc
    ...)
```

The *requirements.txt* file was changed as follows:

```
# sklearn
scikit-learn
```

Then the following command was run again:

```
!python -m pip install -e insightface/recognition
```

In [ ]:
```
!python -m pip install -e insightface/recognition
```

```
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Obtaining file:///~/edds/insightface/recognition
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
  ...
  Building wheel for numpy (setup.py): started
  Building wheel for numpy (setup.py): finished with status 'error'
  Running setup.py clean for numpy
Successfully built easydict
Failed to build numpy
```

```
WARNING: Ignoring invalid distribution - (~\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (~\python310\lib\site-packages)
  WARNING: Ignoring invalid distribution - (~\python310\lib\site-packages)
  WARNING: Ignoring invalid distribution -ip (~\python310\lib\site-packages)
  error: subprocess-exited-with-error

  × python setup.py bdist_wheel did not run successfully.
  │ exit code: 1
  ╰─> [264 lines of output]
      Running from numpy source directory.
  ...
  note: This error originates from a subprocess, and is likely not a problem with pi
p.
  ERROR: Failed cleaning build dir for numpy
ERROR: Could not build wheels for numpy, which is required to install pyproject.toml
-based projects
```

insightface/recognition installation was not successful. The device used for the experiment was a company bound device. Therefore, we were not able to install the *insightface/recognition* package as it reuired dependencies and rights that we did not have. We couldn't overcome this issue. Therefore, we decided to move on to the next step.

Training

In the documentation EXPERIMENT.rst, this section is introduced with the following setup explanation: *We train DPR using 4 V100 GPUs of 32GB, allowing a total batch size of 256 (32 questions * 2 passages each * 4 GPUs). This is crucial because each question uses all passages paired with other questions in the batch as negative examples. Each question is paired with 1 relevant passage and 1 irrelevant passage mined with BM25.* Apperantly, the setup is way above the setup we used for the experiment. Even so, we tried to check the validity of the concept. To do so, we skipped the training part as mentioned, *each question uses all passages paired with other questions in the batch as negative examples*, with our hardware setup, it would take a long time to train the model. Therefore, we decided to move on to use pretrained models.

Following pretrained models are used for the experiment:

- question model: https://huggingface.co/PaulLerner/question_eca_l6_wit_mict
- context/passage model: https://huggingface.co/PaulLerner/context_eca_l6_wit_mict

The EXPERIMENT.rst file is not clear about the usage of the pretrained models. It provides the following command with a description:

As a sanity check, you can check the performance of the models on WIT's test set.

In [ ]:
```
!python -m meerqat.train.trainer test --config=experiments/ict/ilf/config.yaml
```

```
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\train\trainer.py", line 42, in <module>
    main()
  File "~\edds\meerqat\train\trainer.py", line 31, in main
    cli = LightningCLI(
TypeError: LightningCLI.__init__() got an unexpected keyword argument 'description'
```

In [ ]:
```
!python -m meerqat.train.trainer test --config=experiments/ict/eca/config.yaml
```

```
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\train\trainer.py", line 42, in <module>
    main()
  File "~\edds\meerqat\train\trainer.py", line 31, in main
    cli = LightningCLI(
TypeError: LightningCLI.__init__() got an unexpected keyword argument 'description'
```

To utilize the pretrained models, the following models are indicated in the *experiments/ict/eca/config.yaml* file:

```
question_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/question_model_b

    context_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/context_model_be
```

and the following models are indicated in the *experiments/ict/ilf/config.yaml* file:

```
question_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/question_model

    context_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/context_model
```

In [ ]:
```
# Load model directly
from transformers import AutoModel

model = AutoModel.from_pretrained("PaulLerner/context_eca_l6_wit_mict")
```

```
# save the model to a file
model.save_pretrained('experiments/pretrained_models/context_eca_l6_wit_mict')
```

```
pytorch_model.bin:   0%|          | 0.00/439M [00:00<?, ?B/s]
```

```
Some weights of BertModel were not initialized from the model checkpoint at PaulLern
er/context_eca_l6_wit_mict and are newly initialized: ...
  You should probably TRAIN this model on a down-stream task to be able to use it fo
r predictions and inference.
```

In [ ]:
```
# Load model directly
from transformers import AutoModel

model = AutoModel.from_pretrained("PaulLerner/question_eca_l6_wit_mict")

# save the model to a file
model.save_pretrained('experiments/pretrained_models/question_eca_l6_wit_mict')
```

```
config.json:   0%|          | 0.00/932 [00:00<?, ?B/s]
pytorch_model.bin:   0%|          | 0.00/439M [00:00<?, ?B/s]
```

```
Some weights of BertModel were not initialized from the model checkpoint at PaulLern
er/question_eca_l6_wit_mict and are newly initialized: ...
  You should probably TRAIN this model on a down-stream task to be able to use it fo
r predictions and inference.
```

For correcting the path for the pretrained models, the following changes are made to the code: experiements/ict/ilf/config.yaml

```
#question_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/question_model

    #context_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/context_model
    question_model_name_or_path:
experiments/pretrained_models/question_eca_l6_wit_mict
    context_model_name_or_path:
experiments/pretrained_models/context_eca_l6_wit_mict
```

experiements/ict/eca/config.yaml

```
#question_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/question_model_b

    #context_model_name_or_path:
experiments/dpr/triviaqa/lightning_logs/version_0/step=13984/context_model_be

    question_model_name_or_path:
experiments/pretrained_models/question_eca_l6_wit_mict
    context_model_name_or_path:
experiments/pretrained_models/context_eca_l6_wit_mict
```

In [ ]:
```
!python -m meerqat.train.trainer test --config=experiments/ict/ilf/config.yaml
```

```
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\train\trainer.py", line 42, in <module>
    main()
  File "~\edds\meerqat\train\trainer.py", line 31, in main
    cli = LightningCLI(
TypeError: LightningCLI.__init__() got an unexpected keyword argument 'description'
```

The errors do not result from our model path changes but from cli initialization. The *desciption* argument is commented to avoid the error. The following changes are made to the code: experiements/ict/ilf/train.py

```
cli = LightningCLI(
        trainer_class=pl.Trainer,
        # same default as transformers although it is unlikely that the
calls are in the exact same order
        # N. B. called with `workers=True` in LightningCLI
        seed_everything_default=42,
        #description='Main training script based on pytorch lightning.'
    )
```

In [ ]:
```
!python -m meerqat.train.trainer test --config=experiments/ict/eca/config.yaml
```

```
Traceback (most recent call last):
  File "~\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "~\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "~\edds\meerqat\train\trainer.py", line 42, in <module>
    main()
  File "~\edds\meerqat\train\trainer.py", line 31, in main
    cli = LightningCLI(
  File "~\Python310\lib\site-packages\pytorch_lightning\cli.py", line 375, in __init
__
    self.setup_parser(run, main_kwargs, subparser_kwargs)
  File "~\Python310\lib\site-packages\pytorch_lightning\cli.py", line 407, in setup_
parser
    self.parser = self.init_parser(**main_kwargs)
  File "~\Python310\lib\site-packages\pytorch_lightning\cli.py", line 397, in init_p
arser
    parser = LightningArgumentParser(**kwargs)
  File "~\Python310\lib\site-packages\pytorch_lightning\cli.py", line 94, in __init_
_
    raise ModuleNotFoundError(f"{_JSONARGPARSE_SIGNATURES_AVAILABLE}")
ModuleNotFoundError: DistributionNotFound: The 'typeshed-client>=2.1.0; extra == "si
gnatures"' distribution was not found and is required by jsonargparse. HINT: Try run
ning `pip install -U 'jsonargparse[signatures]>=4.26.1'`
```

Another dependency error occured which is not mentioned in the documentation. The following command is used to install the dependency:

In [ ]:
```
!python -m pip install -U jsonargparse[signatures]>=4.26.1
```

```
WARNING: Ignoring invalid distribution - (~\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (~\python310\lib\site-packages)
WARNING: Ignoring invalid distribution - (~\python310\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (~\python310\lib\site-packages)
```

In [ ]:
```
!python -m meerqat.train.trainer test --config=experiments/ict/ilf/config.yaml
```

```
usage: trainer.py [-h] [-c CONFIG] [--print_config[=flags]]
                  {fit,validate,test,predict} ...
error: Validation failed: No action for key "trainer.num_processes" to check its val
ue.
```

We are not clear about the error above.

As far as the documentation is concerned, most of the steps are not clear what are the requirements, if it is already provided, or how to implement it. As a reference in this case, it is not clear if the pretrained models are already provided and executable, even though it is mentioned as a test with pretrained models. For accessing the pretrained models, it is not clear how to implement any model on how to check the performance of the models on WIT's test set.

In general, the authors of the paper did not provide any clear information about the usage of the pretrained models. They provided descriptions to train the models, but hardware constraints did not allow us to train the models so we cannot check it from the trainin side. According to the documentation, a test sequence is provided. In this point, either we did not understand the intended purposes or the already confusing documentation is not as clear as it seems. We have also tried other pretrained models from earlier step, but we could not find a way out. Therefore, we decided to move on to the next step as no test is aplicable from our understanding.

Information Retrieval

The documentation explains this section as follows:

Now that we have a bunch of dense representations, let's see how to retrieve information! Dense IR is done with faiss and sparse IR is done with elasticsearch, both via HF datasets. We'll use IR on both TriviaQA along with the complete Wikipedia (BM25 only) and ViQuAE along with the multimodal Wikipedia.

For installation of the *elasticseach* tool, Elastic documentation is provided. From the link, we have downloaded the *elasticsearch-8.12.0-windows-x86_64* file and extracted it to the *~\elasticsearch-8.12.0* directory. To run the *elasticsearch* tool locally, Run Elastic locally documentation is provided. From the link, it is stated that *To try out Elasticsearch on your own machine, we recommend using Docker and running both Elasticsearch and Kibana.* Since the hardware requirements are high for the experiment, we have been using a company

bound device. Therefore, we were not able to install the *elasticsearch* tool as it reuired dependencies and rights that we did not have. We couldn't overcome this issue. Therefore, we decided conclude the experiment at this point.