Reproducibility review of: Extracting interrogative intents and concepts from geo-analytic questions

Daniel Nüst ID

2020-07-13

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF at https://osf.io/7xrqg/. To cite the report use

Reviewed paper

Summary

The paper code and a sample dataset were published in an anonymous Figshare record at https://figshare.com/s/b3f8b0834ca63b6c5d60 under a Creative Commons BY-NC-ND 4.0 license. I could execute the workflow without errors following the provided instructions. The scripts created a subset of the figures only, *some key figures were not created by the provided data and code*. The authors show good concern for transparency and reproducibility and this *reproduction was partially successful*.

Reproducibility reviewer notes

The paper contains an a Data and Software Availability section and a link to an anonymous Figshare record at https://figshare.com/s/b3f8b0834ca63b6c5d60 published under a CC-BY-NC-ND 4.0 license, which does not match the licensing information in the README ("MIT license").

All scientific reviewers took note of the repository, but did not attempt execution or reproduction. I skimmed the article briefly and then downloaded the archive from Figshare and continued with the succinct `readme.md`. The author kindly reported the expected execution time ("10 hours"), so I created a virtual environment and started the workflow.

The environment uses Python 3.7 (not 3.6 as reported by the authors). The review repository contains a `Pipfile.lock` describing the used environment in detail, and was created with the following commands:

```
pipenv --python 3.7
pipenv install xlsxwriter matplotlib numpy scipy pandas sklearn wordcloud torch allennlp
pipenv lock
pipenv lock -r > requirements.txt
```

Parsing

```
$ pipenv shell

# cd 11407371/GeoParser-AGILE2020/ > Figshare's ZIP, GeoParser-AGILE2020.zip
$ python parse.py
```

Code ran overnight and the final log lines and files created in the directory `parsing_result` are below.

```
[...]
INFO:root:******************************************************
INFO:root:Processing the record number ::: 24559; Currently 100.0% of the MS MARCO is parsed
INFO:allennlp.models.archival:removing temporary unarchived model dir at /tmp/tmpgtqmz3z3
INFO:allennlp.models.archival:removing temporary unarchived model dir at /tmp/tmpnmwgsmfe

$ ll -h parsing_result/
total 23M
drwxrwxr-x 2 daniel daniel 4,0K Jun  5 04:33  ./
drwxrwxr-x 9 daniel daniel 4,0K Dez 23 00:47  ../
-rw-r--r-- 1 daniel daniel 698K Jun  4 23:26  GeoAnQu.json
-rw-r--r-- 1 daniel daniel 209K Jun  4 23:29  GeoQuestion201.json
-rw-r--r-- 1 daniel daniel  22M Jun  5 04:33 'MS MARCO.json'
```

That seems like a successful execution of that script.

Visualisation

```
$ python visualization.py

INFO:allennlp.modules.elmo:Initializing ELMo
```

```
INFO:root:analyzing GeoQuestion201...
INFO:root:analyzing GeoAnQu...
INFO:root:analyzing MS MARCO...
INFO:root:Comparing datasets (random subset: 200, dimension: 17d)...
INFO:root:Comparing datasets (all)...

$ ll -h graphs/
total 19M
drwxrwxr-x 2 daniel daniel 4,0K Jun  5 08:04 ./
drwxrwxr-x 9 daniel daniel 4,0K Dez 23 00:47 ../
-rw-r--r-- 1 daniel daniel 149K Jun  5 08:02 2d_encoding_comparison.png
-rw-r--r-- 1 daniel daniel 224K Jun  5 08:04 2d_encoding_comparison-without-sampling.png
-rw-r--r-- 1 daniel daniel 471K Jun  5 08:02 GeoAnQu_activities.png
-rw-r--r-- 1 daniel daniel 762K Jun  5 08:02 GeoAnQu_objects.png
-rw-r--r-- 1 daniel daniel 710K Jun  5 08:02 GeoAnQu_ointents.png
-rw-r--r-- 1 daniel daniel 744K Jun  5 08:02 GeoAnQu_oqualities.png
-rw-r--r-- 1 daniel daniel 714K Jun  5 08:01 GeoAnQu_pnames.png
-rw-r--r-- 1 daniel daniel 778K Jun  5 08:01 GeoAnQu_ptypes.png
-rw-r--r-- 1 daniel daniel 728K Jun  5 08:02 GeoAnQu_qualities.png
-rw-r--r-- 1 daniel daniel 750K Jun  5 08:02 GeoAnQu_questions.png
-rw-r--r-- 1 daniel daniel 694K Jun  5 08:02 GeoAnQu_situations.png
-rw-r--r-- 1 daniel daniel 667K Jun  5 08:02 GeoAnQu_tintents.png
-rw-r--r-- 1 daniel daniel  95K Jun  5 08:01 GeoQuestion201_activities.png
-rw-r--r-- 1 daniel daniel 619K Jun  5 08:01 GeoQuestion201_objects.png
-rw-r--r-- 1 daniel daniel 340K Jun  5 08:01 GeoQuestion201_ointents.png
-rw-r--r-- 1 daniel daniel 362K Jun  5 08:01 GeoQuestion201_oqualities.png
-rw-r--r-- 1 daniel daniel 713K Jun  5 08:01 GeoQuestion201_pnames.png
-rw-r--r-- 1 daniel daniel 575K Jun  5 08:01 GeoQuestion201_ptypes.png
-rw-r--r-- 1 daniel daniel 478K Jun  5 08:01 GeoQuestion201_qualities.png
-rw-r--r-- 1 daniel daniel 718K Jun  5 08:01 GeoQuestion201_questions.png
-rw-r--r-- 1 daniel daniel 201K Jun  5 08:01 GeoQuestion201_situations.png
-rw-r--r-- 1 daniel daniel 510K Jun  5 08:01 GeoQuestion201_tintents.png
-rw-r--r-- 1 daniel daniel 746K Jun  5 08:02 MSMARCO_activities.png
-rw-r--r-- 1 daniel daniel 777K Jun  5 08:02 MSMARCO_objects.png
-rw-r--r-- 1 daniel daniel 792K Jun  5 08:02 MSMARCO_ointents.png
-rw-r--r-- 1 daniel daniel 740K Jun  5 08:02 MSMARCO_oqualities.png
-rw-r--r-- 1 daniel daniel 782K Jun  5 08:02 MSMARCO_pnames.png
-rw-r--r-- 1 daniel daniel 446K Jun  5 08:02 MSMARCO_ptypes.png
-rw-r--r-- 1 daniel daniel 819K Jun  5 08:02 MSMARCO_qualities.png
-rw-r--r-- 1 daniel daniel 804K Jun  5 08:02 MSMARCO_questions.png
-rw-r--r-- 1 daniel daniel 436K Jun  5 08:02 MSMARCO_situations.png
-rw-r--r-- 1 daniel daniel 426K Jun  5 08:02 MSMARCO_tintents.png
```
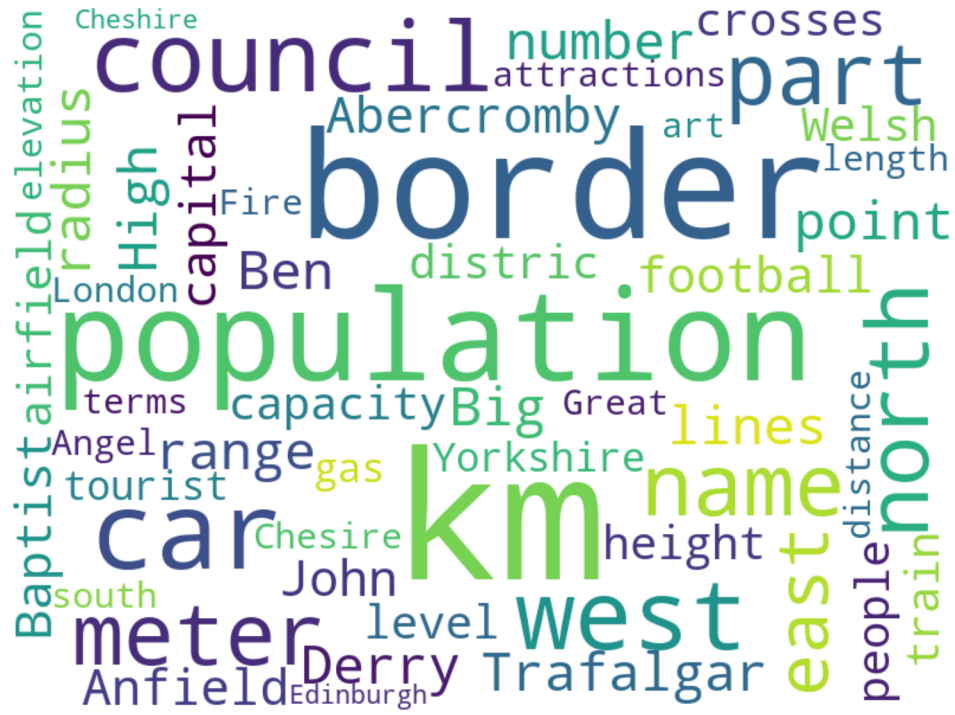
The `graphs` directory now contains a number of plots, some of which I could match to Figures in the paper: `GeoQuestion201_objects.png` to **Figure 6 (a), `GeoQuestion201_situations.png` seems to be Figure 7 (c), but with the additional word "flows", `GeoQuestion201_activities.png` matches Figure 7 (d), `2d_encoding_comparison.png` seems to match Figure 10 in content, but not in the shown data values. For some figures there seems to be no match in the paper, e.g., `2d_encoding_comparison-without-sampling.png`.

This part of the workflow seems reproducible, the differences in Figures are possibly due to randomness effects. Some figures were not included in the reproductions, e.g. Figures 3, 4, 5 (a), 8, 9, 11, 12, 13.
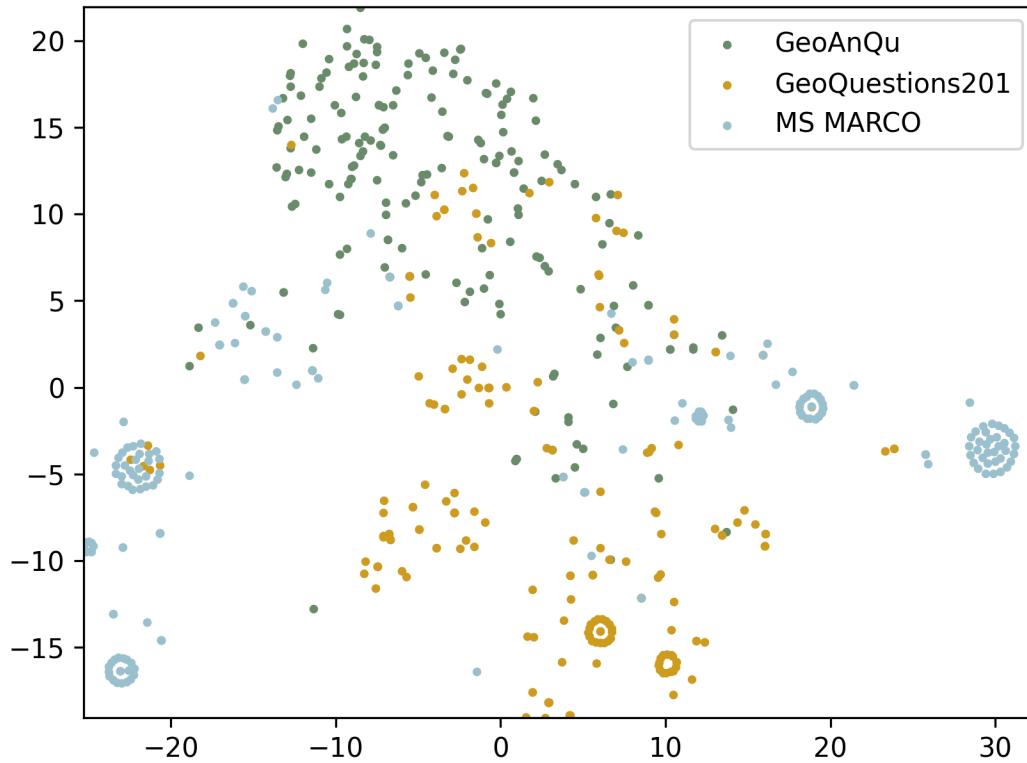
Reproduced Figure 6 (a)

runs

cross

plays

Figure 10

Evaluation script

```
$ cd evaluation
$ python evaluation.py
[...]
$ ll -h
total 180K
drwxrwxr-x 2 daniel daniel 4,0K Jun  5 09:05 ./
drwxrwxr-x 9 daniel daniel 4,0K Dez 23 00:47 ../
-rw-rw-r-- 1 daniel daniel  22K Dez 23 00:47 1.csv
-rw-rw-r-- 1 daniel daniel  21K Dez 23 00:47 2.csv
-rw-rw-r-- 1 daniel daniel  22K Dez 23 00:47 3.csv
-rw-rw-r-- 1 daniel daniel  21K Dez 23 00:47 4.csv
-rw-r--r-- 1 daniel daniel 8,4K Jun  5 09:05 a1-questions.txt
-rw-r--r-- 1 daniel daniel 8,2K Jun  5 09:05 a2-questions.txt
-rw-r--r-- 1 daniel daniel 8,5K Jun  5 09:05 a3-questions.txt
-rw-r--r-- 1 daniel daniel 8,1K Jun  5 09:05 a4-questions.txt
-rw-r--r-- 1 daniel daniel  148 Jun  5 09:05 errors.txt
-rw-rw-r-- 1 daniel daniel  22K Dez 23 00:47 evaluation.py
```

Again, this part of the workflow seems to execute without error, but it's unclear what information these files provide to the article.

Comments to the authors

I briefly skimmed over the script files, mostly to identify the expected outputs. I find the files readable and well formatted. The file names of most data/input/output files are understandable with the background of the paper. The private Figshare project is very well suited for peer review but should be made public and linked to via a DOI for the final publication.

Overall: good job, it was a good experience to reproduce the workflow. I have the following concrete recommendations:

- Reconsider the very restrictive license for your own code and data.

- Better connect created plot names with the figures of the article, e.g., by matching Figure numbers (though I understand that can be tedious) or matching file name and the figure text. Also, consider not creating plots that are not found in the paper at all.

- The list of required libraries is good, you could do even better with a ready-to-use environment spec, e.g., using Pipenv or Conda.

- For higher reproducibility, maybe you can provide files needed within the repository, e.g., when running `parse.py` several logs mention downloading of (model?) files, e.g.,

  ```
  INFO:allennlp.common.file_utils:https://allennlp.s3.amazonaws.com/models/
    fine-grained-ner-model-elmo-2018.12.21.tar.gz not found in cache, downloading to ...
  ```

  which could result in broken workflows if the downloads are not available anymore.

- Document in the README which files are generated by which script.

- Consider setting a seed so that the reproduction creates precisely the same wordcloud.

- There is a function `read_dummy_samples` in `parse.py` > consider providing a synthetic dataset for quicker demonstration of your workflow, together with a small set of expected output files to compare against.