SSI/Harmony - Research Software Healthcheck

Project Title SSI/Harmony

Authorship Steve Crouch

Document Version v1.0

Date 2023-12-14

1 Harmony Deployment & Basic Usage

This section covers deploying and basic use of the software, as well as tools and infrastructure used to build it.

The platform used to test deployment was:

- Mac OS 12.4 Monterey
- Python 3.10.12
- Google Chrome 120.0.6099.71

Observations and suggestions:

U1. The repository README is very informative, and covers many important aspects of deploying and using the software, but may benefit from the following:
    - Consider adding a list of key software features near the top, perhaps as a bullet point list (the README equivalent of an elevator pitch - how would you sell the software?)
    - Many of the listed technical aspects are lacking a narrative that ties them together, and their particular relevance to new users. Consider grouping such content into a more overall structure for the technical aspects, e.g. pre-requisites, getting started (very basic use case), advanced use (covering optional or more advanced usage), frequently asked questions, etc.
    - The intro to using the software would benefit from a stronger step-by-step "Getting Started" narrative from initial installation to a basic example of use
    - There are several ways to use/install/deploy Harmony which is helpful, but it may be difficult to differentiate between them. Perhaps add some brief clarification on what each of these offer
    - Perhaps link to the excellent how harmony works article[1] from the README's description

U2. For someone trying to understand the technical ecosystem of the software, consider adding a brief overview section that briefly covers the key software components used, what they do, and how they fit together

U3. The install video is very engaging and effective in introducing the software, how to install and use it, and key capabilities. It's good at covering the reasoning for each step, and providing clear recaps at key points
    - The example given is particularly effective in showcasing what the software is about, and was able to follow along until the end. Perhaps consider a very simplified version of this could be included in the README within a "Getting Started" section?
    - However, on the test system, encountered some issues with `import harmony`:

[1] https://harmonydata.ac.uk/how-does-harmony-work/

        - `TypeError: issubclass() arg 1 must be a class`: had to downgrade the Python package `typing_extensions` to version 4.5.0 for this to work[2]
        - It didn't download the sentence transformer models as depicted, since sentence transformers weren't available. Had to do `pip install sentence-transformers` first
    - At around ~4:20, some explanations about notebooks were given whilst initialising the notebook, without explaining what the steps were (otherwise, the video was very good at explaining what was being typed whilst doing it)
    - The Jupyter browser text was quite small and difficult to read if the video is viewed in a typically in-window YouTube browser session. In the future, consider using larger browser text (i.e. zoomed in) so it's easier to read in such a setting for those following along. Similarly for the command line text

U4. The use of Google Colab was also an excellent way to showcase the software, step-by-step

- Perhaps consider linking to this also from the video, since essentially this covers a very similar example?
- This could be expanded with explanatory text covering rationale and what's happening at each of the steps for beginners (e.g. the structure of the similarity matrix, etc.)
- Do you need both, since that's more to maintain?
- Was able to complete the Python notebook, but encountered a non-fatal error during `pip install harmonydata`, see Appendix A for error trace. This did not seem to affect the results

Other:

U5. There is a broken link (https://app.harmonydata.ac.uk/) in the README description

[2] Something to do with `pydantic` package? See following for example: https://github.com/langchain-ai/langchain/issues/5113#issuecomment-1558493486

2 Developer Perspective

This section covers the experiences of approaching and using the software from the perspective of a developer.

The platform used to test was:

- Ubuntu 22.04 LTS
- Python 3.10.12
- Java OpenJDK Runtime Environment 11.0.21

Observations and suggestions:

D1. Consider adding a section to the repository README which makes explicit the supported versions of Python, operating systems, etc.

D2. Consider supplying a `requirements.txt` (pip) or `requirements.yml` (Anaconda) file in the root of the repository containing the specific Python package versions for which the system has been developed/tested (e.g. the output from a `pip freeze` or `conda env export` command). This practice will help avoid potential issues with newer libraries, keep package within known working versions, and make it easier for users to set up and use Python virtual environments with specific versions of these packages

- Subsequently, update to newer package versions, retest, and regenerate the requirements file periodically, e.g. during a development cycle and prior to a new release
- Use of virtual environments within software projects a widely accepted practice, so consider introducing use of these within the getting started guide

D3. For new developers, consider having a technical architecture document that covers the structure of the codebase, how the components fit together, and any existing/recommended extension/API points in the code for adding features to the software

D4. Perhaps expand on how to contribute to Harmony by adding a `CONTRIBUTING.md` file with additional detail on how to do this, i.e. to entice people to contribute in particular areas, how to structure issues and pull requests, etc. See the GitHub guide[3] on how you might do this, and for examples

D5. The four automated tests ran successfully, with very solid test coverage (61% of codebase)

- Encountered a deprecation warning from TestConvertExcelXlswriter `DeprecationWarning: np.find_common_type is deprecated. Please use `np.result_type` or `np.promote_types`.` To avoid code

[3] https://docs.github.com/en/communities/setting-up-your-project-for-healthy-contributions/setting-guidelines-for-repository-contributors

breakages when such methods are eventually deprecated, it's best to locate and resolve these
    - Encountered a warning from harmony/parsing/text_extraction/smart_document_parser.py:116:
      ```
      SettingWithCopyWarning:
      A value is trying to be set on a copy of a slice from a DataFrame
      ```
    - The included tests would seem to be "functional" high-level tests as opposed to unit tests, so consider including additional low-level unit tests. This can take some effort, so prioritise critical functions across the codebase and expand later

D6. Managed to run tox to completion successfully
    - Java needed to be installed first (this is mentioned above in the README, but isn't immediately obvious it's needed here)
    - The referenced "Generate distribution files" section isn't present in the README
    - There are a number of deprecation warnings from Tika when pytests are running which should be investigated and removed

D7. The codebase is generally well written, consistent and structured and employs many established best practices and tools
    - Commenting level is generally quite good, although consider adding missing docstrings for functions
    - Some minor improvements to the code style could be achieved using a static code analyser such as Pylint[4] or Flake8[5] (if not already), although consider adding a Pylint/Flake8 config file to the repo to silence long line warning messages
    - Running Bandit[6] (a tool to find common security issues in Python code) over the codebase highlights one high-level and some medium-level potential issues which should be investigated if used as a service. See Appendix B for Bandit output

Other resources:

- Some general SSI tips on managing your open source community effectively

[4] https://pypi.org/project/pylint/
[5] https://flake8.pycqa.org/en/latest/
[6] https://pypi.org/project/bandit/

3 Community Engagement

This section covers aspects related to how the project engages with and builds its user/developer community.

Observations and suggestions

Develop a communication strategy and promote: in order to raise awareness and build a community, promotion is something that needs to happen frequently across a number of communication channels

C1. As a communication principle, consider the potential users and customers as *collaborators* first. The most successful projects avoid "over the wall" solutions and deeply and meaningfully engage their communities. How to do this effectively varies greatly from project to project, e.g. Ersilia Model Hub[7] and JournalTOCS[8] are examples with similar community challenges

C2. Consider building a listed network of outreach contacts across the community if not doing this already, at a number of levels (e.g. national, institutional, community, research group; initially gather from across the partners)

- For Slack, you can make use of existing Slack workspaces such as the UK Research Software Engineers workspace for advertising events and other activities (see the RSE Society website for how to join[9]). As the software community grows, you can create a Harmony Slack workspace (e.g. with general, announce, support, and developer channels)
- Mailing lists are often low-read but also low-effort ways to publicise, so you may want to consider a Mailchimp or similar provider with a subscribe option on the website (e.g. like the SSI[10] - see the bottom of the front page)

C3. The website has many well-written articles; consider cross-posting them to outreach contacts (or just contribute new ones) to increase readership. For example, e.g. the SSI's blog is read by over 20,000/month (contact Selina Aragorn, the SSI Communications Lead and Associate Director of Operations if interested)

- Since community engagement and growth is a key objective, consider always including a "call to action" in the articles or other media appropriate to the post, e.g. getting them to subscribe to a mailing list or Slack workspace, direct email - anything that encourages engagement, and a means to build a contact network
- Encourage (and possibly co-write) relatively short articles from community users about how they use Harmony as a first step, which optionally become case studies later with some expansion

[7] https://www.software.ac.uk/blog/learnings-software-sustainability-health-check
[8] https://www.software.ac.uk/blog/being-business-experiences-research-project
[9] https://society-rse.org/join-us/
[10] https://www.software.ac.uk/

C4. Ensure you have a means to measure readership on the website (e.g. via Google Analytics, website monitoring tools). By coordinating (or staggering slightly) publication dates of posts, cross-posting, and other outreach activities across channels, you can roughly gauge which posts and channels are most effective and are bringing in readers and engagement

C5. Work with users/collaborators to write case studies on adoption of Harmony, and include these on a prominent section of the website, and linked to from the GitHub repository. When considering adopting software, it's very useful for researchers to see and understand how it's being used successfully, particularly when case study use cases match their own. Initial case studies gathered from project partners are similarly effective. The website blog[11] has a couple of these that could be expanded upon and included

Engagement with users/developers

C6. Consider setting up a small, informal, technical test group of "friends and family" (i.e. perhaps 2 or 3 external stakeholders) to test the software as it develops. This group then helps act as a quality gateway to a wider public release
    - You could formalise the group as a Community (or Technical) Advisory Board or similar to recognise and incentivise their involvement (this may make more sense later on, as engagement develops)
    - Ask them to test out new process updates based on draft documentation, e.g. contribution guidelines
    - As a further means to incentivise membership, prioritise working with them (as a group, or individually) to understand their use cases and what needs to improve to make this work best for them, capturing these issues as requirements (e.g. as issues on the Harmony repository)

C7. Researchers are keen to attend training events that increase their knowledge and skills, particularly free online events over a short period of time, e.g.
    - A 'webinar' mini-training event - a couple of hours for a software introduction with a very lightweight "hands on" instructor-led segment to get people started and engaged - takes some time to prepare, but once the preparation is done, re-running such events is relatively low effort
    - With a greater level of community interest, longer full-day training events that delve into more detail and more advanced use, or hackathons that explore how attendee use cases can be realised using the software
    - Once the training preparation and materials has been produced, can reuse and convert these into online self-guiding materials people can access at any time

C8. Attempt to obtain - and encourage - a good balance of feedback from diverse software operating environments (those that are representative of the wider community and intended to be supported) from adopters to help identify and resolve issues across them

[11] https://harmonydata.ac.uk/news-coverage/

C9. For particularly engaged users/developers, consider recruiting them as official project champions, with recognition and perhaps mini project biographies on the website

Development process

C10. Consider short, iterative development cycles, with each resulting in a Release Candidate for community stakeholders to test and provide feedback. This helps the product progress in validated steps towards increasingly mature releases that meet the identified needs of its users
    - Building on this, consider making use of public GitHub milestones to group issues into future release targets, and publicise around that

Other

C11. On the https://harmonydata.ac.uk/ website there are some broken links
    - Main page: for the GitHub "contribute" and "raise an issue links", which point to the incorrect repository
    - The "help us improve harmony"post about workshops[12], a broken "workshops" link

Other resources:

- SSI guide which contains suggestions on supporting open source software

[12] https://harmonydata.ac.uk/sign-up-to-test-harmony/

Appendix A: Error trace from Google Colab install of Harmony

```
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
lida 0.0.10 requires fastapi, which is not installed.
lida 0.0.10 requires kaleido, which is not installed.
lida 0.0.10 requires python-multipart, which is not installed.
lida 0.0.10 requires uvicorn, which is not installed.
llmx 0.0.15a0 requires cohere, which is not installed.
llmx 0.0.15a0 requires openai, which is not installed.
llmx 0.0.15a0 requires tiktoken, which is not installed.
en-core-web-sm 3.6.0 requires spacy<3.7.0,>=3.6.0, but you have spacy
3.5.3 which is incompatible.
google-colab 1.0.0 requires pandas==1.5.3, but you have pandas 2.0.0
which is incompatible.
```

Appendix B: Output from running Bandit security tool over codebase

```
Run started:2023-12-13 13:08:35.508627

Test results:
>> Issue: [B113:request_without_timeout] Requests call without timeout
   Severity: Medium   Confidence: Low
   CWE: CWE-400 (https://cwe.mitre.org/data/definitions/400.html)
   More Info:
https://bandit.readthedocs.io/en/1.7.6/plugins/b113_request_without_ti
meout.html
   Location:
src/harmony/parsing/text_extraction/ensemble_named_entity_recogniser.p
y:83:19
82                "HARMONY_NER_ENDPOINT") != "":
83            response = requests.get(
84                os.environ.get("HARMONY_NER_ENDPOINT"), json={"text":
json.dumps([page_text])})
85            doc_bin = DocBin().from_bytes(response.content)

--------------------------------------------------
>> Issue: [B113:request_without_timeout] Requests call without timeout
   Severity: Medium   Confidence: Low
   CWE: CWE-400 (https://cwe.mitre.org/data/definitions/400.html)
   More Info:
https://bandit.readthedocs.io/en/1.7.6/plugins/b113_request_without_ti
meout.html
   Location:
src/harmony/parsing/text_extraction/ensemble_named_entity_recogniser.p
y:167:19
166       if os.environ.get("HARMONY_CLASSIFIER_ENDPOINT") is not None
and os.environ.get("HARMONY_CLASSIFIER_ENDPOINT") != "":
167           response = requests.get(
168               os.environ.get("HARMONY_CLASSIFIER_ENDPOINT"),
json={"text": json.dumps([q.question_text for q in questions])})
169           doc_bin = DocBin().from_bytes(response.content)

--------------------------------------------------
>> Issue: [B108:hardcoded_tmp_directory] Probable insecure usage of
temp file/directory.
```

```
   Severity: Medium   Confidence: Medium
   CWE: CWE-377 (https://cwe.mitre.org/data/definitions/377.html)
   More Info:
https://bandit.readthedocs.io/en/1.7.6/plugins/b108_hardcoded_tmp_dire
ctory.html
   Location: src/harmony/parsing/util/camelot_wrapper.py:45:14
44
45        tmpfile = "/tmp/" + uuid.uuid4().hex + ".pdf"
46        with open(tmpfile, "wb") as f:

--------------------------------------------------
>> Issue: [B410:blacklist] Using html to parse untrusted XML data is
known to be vulnerable to XML attacks. Replace html with the
equivalent defusedxml package.
   Severity: Low   Confidence: High
   CWE: CWE-20 (https://cwe.mitre.org/data/definitions/20.html)
   More Info:
https://bandit.readthedocs.io/en/1.7.6/blacklists/blacklist_imports.ht
ml#b410-import-lxml
   Location: src/harmony/parsing/util/tika_wrapper.py:31:0
30
31    from lxml import html
32    from tika import parser

--------------------------------------------------
>> Issue: [B202:tarfile_unsafe_members] tarfile.extractall used
without any validation. Please check and discard dangerous members.
   Severity: High   Confidence: High
   CWE: CWE-22 (https://cwe.mitre.org/data/definitions/22.html)
   More Info:
https://bandit.readthedocs.io/en/1.7.6/plugins/b202_tarfile_unsafe_mem
bers.html
   Location: src/harmony/util/model_downloader.py:81:4
80        file = tarfile.open(tmpfile)
81        file.extractall(local_path)
82

--------------------------------------------------
```