Hyperreal Enterprises: Roadmap

Joe Corneli

October 18, 2020

Abstract

This document can be thought of as an informal outline of a "*tactic state*". On a technical level, the upstream source for this material is an Org Roam graph. The "Wiki" section contains instructions for accessing the material and generating derived formats, such as the Org Agenda.

Contents

1 Hyperreal Enterprises: Roadmap

1.1 Preface

This document synthesises a Roadmap (and perhaps also a Product Roadmap) for Hyperreal Enterprises, Ltd. The Roadmap is being written inside Org Roam (an Emacs package), and shared via Git on repo.or.cz.

This document can be thought of and edited as a Wiki, from which other downstream formats can be derived. A narrative introduction follows. To skip it, browse ahead to Top.

1.2 Introduction

1.2.1 Applying AI to technical fields is a huge opportunity.

It's striking that in computer programming work, *collective intelligence* is used almost everywhere, but artificial intelligence is used almost nowhere. AI can kick butt at Chess and Go, but university courses are still taught by human professors. Surely, AI that could write code at a human level would be a valuable thing...

1.2.2 But where's the human-level AI for coding?

A research agenda around "AI for programming" was already spelled out by Alan Turing in the late 1940s. It is only relatively recently that we have massive amounts of relevant data to work with. Code completion and code generation tools are some of the "low hanging fruit" of this domain. We think that something much more substantial is around the corner.

1.2.3 We are looking at this problem as an informal open source R&D collective.

Our team brings together applied experience in Natural Language Processing (NLP), computational modelling, online communities, physics, biology, mathematics, and statistics. We've been getting together for daily coffee chats, and sharing information and skills with each other in online seminars. Sometimes we bring in guest speakers. We hope as time goes by our chats will coalesce into notes, blog posts, papers, and working prototypes. We aren't promising a schedule of deliverables — because we're all volunteers — and we're doing this for fun and interest. But we do plan to share what we're learning as we go along!

1.2.4 Broadly, the steps we have in mind go from data, to models, to AI agents.

We plan to document our progress (or lack thereof) on this blog. As a very rough outline, this is what we expect to look at:

- We plan to use contemporary information extraction methods to derive computationally meaningful material from Stack Exchange Q&A, Github Issues, and programmers' discussions, along with code. To do this, we will combine general purpose language models, like BERT, and a knowledge graph approach.

- We plan to use category theoretic methods as a glue that can hold together a range of computational models, including models of programs and the process of computer programming. Monocl is an existing process modelling language that has been used to create an abstraction layer over a collection of data science programs. We plan to generalise this.

- Ultimately, we plan to install these models in computational agents who can then "converse with each other to sharpen their wits," as Turing anticipated, mirroring contemporary developments in self-play. However, the specific design of the agents remains an open issue at the moment! We plan to explore techniques from Bayesian learning, logic programming, and reinforcement learning.

1.2.5 Reflection is part of the process.

We're interested in understanding human behaviour as well as technical topics: that goes for our own behaviour in particular. We plan to post to our blog at least monthly — as long as we keep our discussion running — using it as a place to reflect on how things are going. Although we're grappling with some weighty topics, success is not all or nothing! Writing here will mark our progress and be useful in their own right (e.g., blog posts can feed into research papers or tools).

Alongside setting up a blog and drafting an initial anouncement (voilà!), we've updated the website of our affiliated UK-based company Hyperreal Enterprises to match the outlines of what we've described here. As mentioned above, are having regular meetings, which we'll record if they look likely to be interesting to a wider audience. We've got nice internal documentation going, via Org Roam, from wich various derived formats are produced, including a public wiki generated by Firn. (We recently sent an abstract to EmacsConf 2020 to talk about the various other Emacs-based tools we're using!)

Some of our seminars so far have explored an Open Information Extraction (with several demo notebooks), and looked at the early parts of Michael Betancourt's "Towards a principled Bayesian workflow".

Active channels on our Discord server include:

- #emacs-cloud-hypernotebooks

- #how-to-design-programs

- #text-analysis

- #model-construction

- #knowledge-graph

We are creating AI tools that will process open source information and build on these in applications such as automated programming.

2   Top

2.1   Motivation: For the sake of advancing AI

We are doing this R&D work partly to make demonstrations of more advanced AI systems. We expect that our stance on AI will not necessarily be a popular one. But this is an important "minor" strand of AI research dating back to Alan Turing:

> "As time goes on the [computer] itself will take over the functions both of [programmers] and of [users]...The [programmers] are liable to get replaced because as soon as any technique becomes at all stereotyped it becomes possible to devise a system of instruction tables which will enable the electronic computer to do it for itself. It may happen however that the [programmers] will refuse to do this. They may be unwilling to let their jobs be stolen from them in this way. In that case they would surround the whole of their work with mystery and make excuses, couched in well chosen gibberish, whenever any dangerous suggestions were made."
> – Alan Turing, 1947.

On average, advanced AI would bring in new ways of working, and would facilitate broad access to high-quality training. This agenda could serve to focus the mind of technical workers, but not many are pursuing it presently.

2.2   Motivation: Technical experiments become easier

Even in the present time, without relying on any speculative AI futures to magically appear, we can benefit from pursuing the agenda above. Accordingly, we are doing some applied work with existing software that will give us a set of further tools and levers to work with.

2.3   Representative Prior Work

2.3.1   PlanetMath

PlanetMath users created a reasonably large informal mathematical knowledge base together. On the way, we came up with several technical demos and sketched possible previews for upcoming features. One possible direction of work we looked at would be to focus on building a comprehensive category theory knowledge base.

2.3.2   Modelling the way mathematics is actually done

In this paper, we talked about how mathematics is situated somewhere in between 'games' and 'storytelling' in its complexity. We proposed to build computational models of informal mathematical reasoning. Subsequent work continued on in this direction, using ideas from dialogue games and argumentation theory more broadly.

2.4   A sketch of a plan

So, having gotten together around these ideas, we're having online chat, frequent short meetings. We've talked about maintaining a blog that would describe what we're learning and developing. So, roughly speaking, we will try to develop a curriculum through the blog. We also have this wiki, that any of us can edit, which we can use as a staging ground for more developed blog posts. Our thought was that blog posts might move in the direction of more developed outputs, whether products or research papers. We want to use some ideas adapted from Scrum to build a shared awareness of what's going on. However, we want to be careful not to become "managerial" since everyone is currently here as a volunteer, working on topics of his or her own interest. We want to provide mutual support and fun. Reflection, whether in writing, or by recording and listening again to conversations, should help with that. We are not constraining things to come out in a purely structured curriculum, or any other form of product development. "Users" and "customers" may appear as we release things we are happy with and expand our little community.

> "Rousseau says, someone who has been properly educated will be engaged in society, but relate to his or her fellow citizens in a natural way. ... We naturally look after our own preservation and interests. By contrast, *amour-propre* is an unnatural self-love that is essentially relational. ... Thus, *amour-propre* can contribute positively to human freedom and even virtue. Nevertheless, *amour-propre* is also extremely dangerous because it is so easily corruptible. ... In its corrupted form, *amour-propre* is the source of vice and misery, and results in human beings basing their own self worth on their feeling of superiority over others." — IEP

2.5   A possible formulation: short correlated sprints as opposed to random behaviour

"Two people working together 4 hours a week for two weeks" could serve as an approximate unit of work. Once we have amassed a few outputs from this kind of effort, we will have some evidence of the kinds of things that we can realistically achieve. So far, our workflow has been more based on solo activities and informal conversations, but short robust team-ups continue to be an option!

> Hypothetical conversation: *In my next post I want to integrate something that I learned from you about PL. I want to drive in the direction of synthesis, as hard as I know how to right now. This depends on everyone having free time to invest in this. Start a blog where we think about what's the overlap in terms of learning?*

3   Why not what

Our purpose:

- We want to make the knowledge economy accessible to everyone.

- Our long-term vision is computational intelligence based on collective intelligence.

3.1 Teach arbitrary coding

This would be an abstraction over teaching basic programming and knowledge graphs.

3.1.1 Feature: Production system

We've started to build a simple production system that can be used to detect errors in subtraction (reimplementing some classic work). We were thinking that something similar could be used to detect other kinds of errors (so, for debugging, teaching), and to support other kinds of reasoning processes (e.g., turning Q's into A's in a question-answering system).

We previously did a little exploratory work, with similar intent, using polygraphs as input, in the workshop paper *Modelling the Way Mathematics Is Actually Done*.

1. Demo application: Reimplementing classic rules to model subtraction

    We looked at a classic paper about "subtraction on Mars" and it seems that reimplementing it might be the best way to go.

3.2 How to Design Programs

We were thinking of *How to Design Programs* (HtDP) as a potential basis for this work. We would want to respect category theoretic concepts in the presentation. We would expect to find analogues in settings like Bayesian modelling.

We could proceed by looking at relationships with argumentation theory, thinking about how to do this in a theoretically consistent way. Once we have a definition of the programming language we're going to use, we can then do argumentation over that.

Another strategy would be to develop a DSL for HtDP ideas, which we could then reuseq to generate patterns for learning how to design various structures (say, web pages or probabilistic programs). To do this well you'd need ways to express 'recipes'. For example, an MVP might be based on representing HtDP-style recipes using sequent calculi for session types. These represent interactive protocols.

You'd use cut-elimination to have two players interact (using something like the Lakatos Game diagram). But what formalism would you use? E.g., *geometry of interaction in linear logic* has been used for this kind of thing, but could it be used here? With a suitable formalism in place we would then imagine that a computer programming agent would just follow the "Lakatos Game" style HtDP script. So, this would contribute to the development of agent models for programming and program-related Q&A.

3.2.1 Related work

- General theory-informed algorithms (e.g., apply category theory to scientific models).

- K framework: Have transformations for any language you define in it.

- HtDP is similar applied to programming teaching. Start with PL theory and then find universal things.

- How can we define statistics in a general way and then derive things from it? (E.g., Anglican probabilistic programming?)

4 Construct, critique, improve models of the creative process

We want tools and processes for working with models, with a particular emphasis on improved models of the creative process. The reason for this emphasis is that if we have good models of the creative process, including the modelling process, we can then apply them to a wide range of problems! This prompts reflection on the infrastructure and tools that we are actually using.

4.1 Emacs Hyper Notebook

We are developing a better way to do "Jupyter notebooks" using Emacs. This recovers some of the Research Collab ideas developed by Aaron Krowne. It should integrate features such as writing and task management (e.g., *Org*) Program evaluation (e.g., *Maxima*), Typesetting and presentation (e.g., slides via *LaTeX*), and navigation (e.g., *Org Roam* for displaying topics as a graph). We should be clear that the various technologies used are slot-fillers and they might be replaced with other things, or augmented (e.g., *Lean* for formal verification of some of the above?). A useful input to this process would be implementation of examples without integration. This can then be redone in a more integrated fashion.

An integration using existing technologies will have limitations, once we have this demos then we will see some of the gaps and how more advanced tech could be useful. (For example, Ray's work with Gerschom could turn out to be useful here.)

1. Some quick thoughts

   - If it was sitting inside a web container, then maybe it's a quickstart thing that comes in a user friendly form.
   - Maybe add menu-bar items to make it look like a web browser.
   - Emacs Maxima interface, we might build on it — for Monday 12th October, a quick "15 minutes" talk to catch up
   - ... possible deliverable for later on: a NIST talk?

4.1.1 **TODO** Figure out subtasks to deliver for EmacsConf   JOE:RAY:CAMERON

4.1.2 **TODO** Figure out how EHN relates to other projects   JOE:RAY:CAMERON

4.1.3 **TODO** Keep testing crdt.el and lockstep.el   JOE:RAY

4.1.4 **TODO** Could Emacsconf talk become a blog post?   JOE:RAY:CAMERON

(Notice that with crdt, typing can go on inside folded nodes! Qiantan is thinking about a mode to make overlays shareable, which would change things a bit.)

4.1.5 Partial prototypes

How far can we go... Through Roam? (We could at least talk to Connor about Roam on Twitter?) Through Jupyter? Foam? Glamorous Toolkit? Can we integrate what we're building with existing tools like these? Do Lenses or other kinds of ACT machinery help with this at all? Would our system potentially play a role as a universal backend?

4.1.6 Feature: Arxana 2020

Revisit Arxana and turn it into something that we can actually use. This is rather closely related to the use of "knowledge graph" formulations we've been discussing, since Arxana allows us to combine writing with knowledge representations. In our last round of work with Arxana, we left off at the point of integrating logic programming into the system.

Technology like this could be used to build simple demos (e.g., Emacs in the browser, running Org Mode). We've noticed some other related tools as well, like Organice and CodiOrg that could provide alternative interfaces.

- exp2exp/notebooks: This is a Docker configuration for running jupyter with multiple kernels on Arch Linux.

- emacsclient Options - GNU Emacs Manual

- butlerx/wetty: Terminal in browser over http/https. (Ajaxterm/Anyterm alternative, but much better)

- xtermjs/xterm.js: A terminal for the web

- https://twitter.com/cianbutlerx

- tslo922/ttyd: Share your terminal over the web

- yudai/gotty: Share your terminal as a web application

- butlerx/wetty - Docker Hub

- Setup Web Terminal using Wetty Docker Image | by Chairat Onyaem (Par) | Medium

- krishnasrinivas/wetty - Docker Hub

4.1.8   Other related work

- James Fairbanks (relate this to Betancourt).

4.1.9   Testing

```
(def a 2)
```

```
#'user/a
```

```
a
```

```
2
```

```
(range 10)
(def a 1)
```

```
| (0 1 2 3 4 5 6 7 8 9) |
| #'user/a              |
```

```
a
```

```
1
```

Configuration of the `src` block might look like:

```
maxima :process :backend maxima :results output org
```

- `:process` stands in for `:session` now as an alternative

- `org-babel-execute-src-block` this is what will be called

- hang into the `lang` parameter of the above function, but override using `:process`

```
100*9
```

```
| value | 900 |
```

We had a short problem with this:

```
1+1;
(error "No such language mode: nil-mode")
...
```

4.1.11   Backends

1. jupyter The jupyter backend works well locally but suffers from a bug when run via tramp. See emacs-jupyter remote debugging

2. ob-streams

   This is work in progress, with some sample content above.

4.1.12   Future work

- Extending to VS Code? Would people who use VS code even want this kind of interaction? Maybe VS Code is better for quick visualisations?

4.2   emacs-jupyter remote debugging

4.2.1   Set up container on gcp

1. gcp configuration

   You may like to run `gcloud auth login` (auth login docs). This is an interactive process that launches oauth for your google account in the web browser so I think it is best to do it from a terminal though it may be possible to run it in org-babel.

```
gcloud config configurations list
```

```
gcloud config configurations describe quarere
```

```
is_active: true
name: quarere
properties:
  compute:
    region: us-central1
    zone: us-central1-f
  core:
    account: camrn86@gmail.com
    project: quarere
```

2. launch container image

   Deploy a vm based on the container `cameronraysmith/notebooks:latest`.

```
gcloud compute instances create-with-container notebooks-vm \
    --container-image registry.hub.docker.com/cameronraysmith/notebooks:latest \
    --container-restart-policy on-failure \
    --container-privileged \
    --container-stdin \
    --container-tty \
  --container-mount-host-path mount-path=/home/jupyter,host-path=/tmp,mode=rw \
    --machine-type n1-standard-1 \
    --boot-disk-size 50GB \
    --preemptible
```

   Setup ssh with your new instance

```
gcloud compute config-ssh
cat ~/.ssh/config | grep "Host notebooks"

You should now be able to use ssh/scp with your instances.
For example, try running:
ssh notebooks-vm.us-central1-f.quarere
Host notebooks-vm.us-central1-f.quarere
```

   You can `ssh` into the host machine or the container using the various commands below.

```
gcloud compute ssh notebooks-vm # into host machine
ssh notebooks-vm.us-central1-f.quarere docker ps -aqf "name=klt-notebooks-
vm-cjme" # check the container ID
gcloud compute ssh notebooks-vm --container klt-notebooks-vm-cjme # use gcloud ssh with -
-dry-run to print the command
ssh -t notebooks-vm.us-central1-f.quarere -- sudo docker exec -it klt-notebooks-
vm-cjme /bin/sh # this takes you directly into the container
```

   Of course you can stop and start the machine with

```
gcloud compute instances stop notebooks-vm
gcloud compute instances start notebooks-vm
```

4.2.2   Startup the cloud vm running our container of interest

1. Setup remote container host machine

   We already setup the container named `notebooks-vm` so all we need to do to begin with is to start it up.

```
gcloud compute instances start notebooks-vm
```

   Check that our instance is indeed running

```
gcloud compute instances list
```

```
NAME         ZONE         MACHINE_TYPE  PREEMPTIBLE INTERNAL_IP EXTERNAL_IP  STATUS
notebooks    us-central1-c n1-standard-1              10.128.0.22             TERMINATED
notebooks-vm us-central1-f n1-standard-1 true         10.128.0.26 35.223.72.41 RUNNING
```

Make sure the correct ip address is entered into our `.ssh/config` file.

```
gcloud compute config-ssh
```

```
You should now be able to use ssh/scp with your instances.
For example, try running:

  $ ssh notebooks-vm.us-central1-f.quarere
```

Inspect the IP address we find in our `.ssh/config` file

```
grep HostName ~/.ssh/config
```

```
HostName 35.224.59.240
HostName 35.223.72.41
```

(a) Execute commands on the remote container host machine

```
hostname --long
```

```
$ notebooks-vm.us-central1-f.c.quarere.internal
```

```
docker container ls
```

```
CONTAINER ID      IMAGE                                                    COMMAND          (
7e5e8974adc8      registry.hub.docker.com/cameronraysmith/notebooks:latest         "/
c 'jupyter…"   About a minute ago   Up About a minute              klt-
notebooks-vm-cjme
5a480b60af52           gcr.io/stackdriver-agents/stackdriver-logging-
agent:0.2-1.5.33-1-1  "/entrypoint.sh /usr…"   About a minute ago   Up About a minute
logging-agent
```

```
docker container ls
```

```
$ CONTAINER ID      IMAGE                   COMMAND          CREATED         STATUS
caadc9a126bb           gcr.io/inverting-proxy/agent        "/bin/sh -
c '/opt/bi…"  2 hours ago     Up 2 hours              proxy-
agent
8080/tcp   payload-container
```

4.2.3   Run shell commands on the remote container

To switch between two available configurations, choose one of the lines below to copy to the `:PROPER-TIES:` drawer for this section.

```
:header-args: :results output verbatim replace :session notebookscontainer :dir /ssh:notebooks-
vm.us-central1-f.quarere|docker:klt-notebooks-vm-cjme: :exports both :eval never-
export
:header-args: :results output verbatim replace :session notebookscontainer :dir /ssh:notebooks.u
central1-c.quarere|docker:payload-container: :exports both :eval never-export
```

In order to connect to the remote host followed by the docker container we specify the directory as `ssh:notebooks-vm` (including the extra details we got from `gcloud compute ssh-config`) followed by `docker:containername` where we got the container name from running `docker container ls` on the remote machine.

```
echo $JUPYTER_PATH
```

```
$
```

```
head -3 /proc/self/cgroup
```

```
$ 12:blkio:/docker/0313f41d52ad7945b5f9687efc0d1e3431e531eb7a29c8d7eecf5fddcbef0f93
11:net_cls,net_prio:/docker/0313f41d52ad7945b5f9687efc0d1e3431e531eb7a29c8d7eecf5fddcbef0f93
10:hugetlb:/docker/0313f41d52ad7945b5f9687efc0d1e3431e531eb7a29c8d7eecf5fddcbef0f93
```

Check the working directory and the list of jupyter kernels

```
(push "-e" docker-tramp-docker-options)
(push "-e" "JUPYTER_PATH=/home/jovyan/.local/share/jupyter:/usr/local/share/jupyter:/usr/share
tramp-docker-options)
```

```
echo $JUPYTER_PATH
```

```
jupyter kernelspec list
```

```
Available kernels:
  ir          /home/jovyan/.local/share/jupyter/kernels/ir
  julia-1.5   /home/jovyan/.local/share/jupyter/kernels/julia-1.5
  maxima      /home/jovyan/.local/share/jupyter/kernels/maxima
  python3     /usr/share/jupyter/kernels/python3
```

If you try to make use of an existing session on the docker container to run one of the `emacs-jupyter` kernels, you find that there is a different usage of the TRAMP remote path specification in the `:dir` property for the `sh` language of babel and with the `:session` property in the `emacs-jupyter` *language* of babel. This is the error I got the first time I tried this with the TRAMP remote path specification in `:dir`:

```
: FileNotFoundErrorTraceback (most recent call last)
: <ipython-input-1-d4b8d99aef95> in <module>
:       1 import os
:       2 __JUPY_saved_dir = os.getcwd()
: ----> 3 os.chdir("/ssh:notebooks-vm.us-central1-f.quarere|docker:klt-notebooks-
vm-cjme:/home/jovyan/")
:       4 try:
:       5     get_ipython().run_cell("""x = 'foo'
:
: FileNotFoundError: [Errno 2] No such file or directory: '/ssh:notebooks-vm.us-
central1-f.quarere|docker:klt-notebooks-vm-cjme:/home/jovyan/'
```

1. Run python session on the remote container The default properties that should apply to this section are

```
:header-args: :results output verbatim replace :session notebookscontainer-
python :dir /ssh:notebooks-vm.us-central1-f.quarere|docker:klt-notebooks-vm-
cjme:/home/jovyan/  :exports both  :eval never-export
```

In order to connect to the remote host followed by the docker container we specify the directory as `ssh:notebooks-vm` (including the extra details we got from `gcloud compute ssh-config`) followed by a `|` and then `docker:containername` where we got the container name from running `docker container ls` on the remote machine.

```
x = 'foo'
y = 'bar'
print(x + ' ' + y)

foo bar


x = 1 + 1
print(x)

2
```

4.2.4 BUG: Run a jupyter kernel in a remote container

1. properties To switch between two available configurations, choose one of the lines below to copy to the `:PROPERTIES:` drawer for this section.

```
:header-args: :results output verbatim replace :session /ssh:notebooks-vm.us-
central1-f.quarere|docker:klt-notebooks-vm-cjme:notebooks01 :exports both :eval never-
export
:header-args: :results output verbatim replace :session /ssh:notebooks.us-
central1-c.quarere|docker:payload-container:notebooks01 :exports both :eval never-
export
```

2. test code

```
x = 'foo'
y = 'bar'
x + ' ' + y
```

There is a problem whereby the `:dir` property is being passed along to jupyter as if it were a file. It looks like the intention may be to pass the TRAMP parameters to `:session` rather than `:dir` in the case of `emacs-jupyter`.

Here there is a bug that has been reported in issue 191 of emacs-jupyter.

```
executing Jupyter-Python code block...
jupyter-start-kernel: default-directory = /ssh:cloudmachine|docker:containeroncloudmachine
jupyter-start-kernel: Starting process with args "/bin/python3 -c from jupyter_client.kerne
-kernel=python3"
```

```
Tramp: Opening connection for containeroncloudmachine using docker...
Tramp: Sending command 'exec ssh -q    -e none cloudmachine'
Tramp: Waiting for prompts from remote shell...done
Tramp: Found remote shell prompt on 'cloudmachine'
Tramp: Sending command 'exec docker  exec -it  containeroncloudmachine sh'
Tramp: Waiting for prompts from remote shell...done
Tramp: Found remote shell prompt on 'containeroncloudmachine'
Tramp: Opening connection for containeroncloudmachine using docker...done
Launching python3 kernel process...done
Tramp: Inserting '/ssh:cloudmachine|docker:containeroncloudmachine:/home/jovyan/.local/sh
fc5b0ea7-f553-4725-aa59-32829d356665.json'...
Tramp: Encoding remote file '/ssh:cloudmachine|docker:containeroncloudmachine:/home/jovyan
fc5b0ea7-f553-4725-aa59-32829d356665.json' with 'base64 <%s'...done
Tramp: Decoding local file '/var/folders/1d/wtzfcz5s4x98nbkdx9g5ss3c0000gn/T/tramp.krOJmR.
decode-region'...done
Tramp: Inserting '/ssh:cloudmachine|docker:containeroncloudmachine:/home/jovyan/.local/sh
fc5b0ea7-f553-4725-aa59-32829d356665.json'...done
SENDING: :kernel-info-request ae928b51-f755-441e-a250-8a08c58d734d nil
SENT: (:shell ae928b51-f755-441e-a250-8a08c58d734d)
Requesting kernel info...done
jupyter-kernel-info: Kernel did not respond to kernel-info request
```

There is a suggestion from arthurcgusmao in another issue stating one needs to set the `JUPYTER_PATH` environment variable to resolve the `Kernel did not respond to kernel-info request` issue.

It is simple to set the `JUPYTER_PATH` environment variable via tramp

```
(add-to-list 'tramp-remote-process-environment "JUPYTER_PATH=/home/jovyan/.local/share/ju
```

however, this does not resolve the issue.

I originally tried to set the environment variable by passing a parameter to docker, but this did not work properly in the sense that if you check the value from inside the container it does not appear to be set despite what appears to be the appropriate docker flag for doing so.

```
(push "-e" "JUPYTER_PATH=/home/jovyan/.local/share/jupyter:/usr/local/share/jupyter:/usr/
tramp-docker-options)
(setq docker-tramp-docker-options
    '("-e" "JUPYTER_PATH=/home/jovyan/.local/share/jupyter:/usr/local/share/jupyter:/usr/
```

3. Debugging `jupyter-kernel-info`

   `jupyter-kernel-info` is the function from which the error `Kernel did not respond to kernel-info request` arises (see line 2066 of jupyter-client.el).

   The stack trace for `jupyter-kernel-info`

```
Debugger entered--entering a function:
jupyter-kernel-info(#<jupyter-org-client jupyter-org-client-1fe73d7aa114>)
jupyter--error-if-no-kernel-info(#<jupyter-org-client jupyter-org-client-1fe73d7aa114>)
jupyter-start-new-kernel("julia-1.5" jupyter-org-client)
jupyter-run-repl("julia-1.5" nil nil jupyter-org-client)
```

14

```
#f(compiled-function (session kernel) "Initiate a client connected to a remote kernel proces
babel-jupyter-remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt...
repl-p nil) "julia-1.5")
apply(#f(compiled-function (session kernel) "Initiate a client connected to a remote kernel
babel-jupyter-remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt...
repl-p nil) "julia-1.5"))
#f(compiled-function (&rest args) #<bytecode 0x1fe743520a15>)(#s(org-babel-
jupyter-remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..." :con
repl-p nil) "julia-1.5")
apply(#f(compiled-function (&rest args) #<bytecode 0x1fe743520a15>) (#s(org-
babel-jupyter-remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt...
repl-p nil) "julia-1.5"))
#f(compiled-function (&rest cnm-args) #<bytecode 0x1fe7430d03fd>)()
#f(compiled-function (cl--cnm session kernel) "Rename the returned client's REPL buffer to i
include-other-output' to nil for the session so\nthat output produced by other clients do not
function (&rest cnm-args) #<bytecode 0x1fe7430d03fd>) #s(org-babel-jupyter-
remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..." :connect-
repl-p nil) "julia-1.5")
apply(#f(compiled-function (cl--cnm session kernel) "Rename the returned client's REPL buffe
include-other-output' to nil for the session so\nthat output produced by other clients do not
function (&rest cnm-args) #<bytecode 0x1fe7430d03fd>) (#s(org-babel-jupyter-
remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..." :connect-
repl-p nil) "julia-1.5"))
#f(compiled-function (&rest args) #<bytecode 0x1fe743520a41>)(#s(org-babel-
jupyter-remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..." :con
repl-p nil) "julia-1.5")
apply(#f(compiled-function (&rest args) #<bytecode 0x1fe743520a41>) #s(org-
babel-jupyter-remote-session :name "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt...
repl-p nil) "julia-1.5")
org-babel-jupyter-initiate-client(#s(org-babel-jupyter-remote-session :name "/ssh:noteboo
vm.us-central1-f.quarere|docker:klt..." :connect-repl-p nil) "julia-1.5")
org-babel-jupyter-initiate-session-by-key("/ssh:notebooks-vm.us-central1-f.quarere|docke
names) (:rowname-names) (:result-params "replace") (:result-type . value) (:results . "repla
export") (:async . "no") (:session . "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..
1.5")))
#f(compiled-function (&optional session params) "Initialize a Jupyter SESSION according to F
vm.us-central1-f.quarere|docker:klt..." ((:colname-names) (:rowname-names) (:result-
params "replace") (:result-type . value) (:results . "replace") (:exports . "both") (:cache
export") (:async . "no") (:session . "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..
1.5")))
apply(#f(compiled-function (&optional session params) "Initialize a Jupyter SESSION accordi
vm.us-central1-f.quarere|docker:klt..." ((:colname-names) (:rowname-names) (:result-
params "replace") (:result-type . value) (:results . "replace") (:exports . "both") (:cache
export") (:async . "no") (:session . "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..
1.5"))))
org-babel-jupyter-initiate-session("/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..
names) (:rowname-names) (:result-params "replace") (:result-type . value) (:results . "repla
export") (:async . "no") (:session . "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..
1.5")))
org-babel-execute:jupyter-julia("x = \"foo\"\ny = \"bar\"\nprintln(x)\nprintln(y)" ((:coln
names) (:rowname-names) (:result-params "replace") (:result-type . value) (:results . "repla
```

export") (:async . "no") (:session . "/ssh:notebooks-vm.us-central1-f.quarere|docker:klt..
1.5")))
#f(compiled-function (&optional arg info params) "Execute the current source code block.\nIr
execution even if an existing\nresult cached in the buffer would otherwise have been returnec
babel-get-src-block-info'.\n\nOptionally supply a value for PARAMS which will be merged with
ob-async-org-babel-execute-src-block(#f(compiled-function (&optional arg info params) "Exe
execution even if an existing\nresult cached in the buffer would otherwise have been returnec
babel-get-src-block-info'.\n\nOptionally supply a value for PARAMS which will be merged with
apply(ob-async-org-babel-execute-src-block #f(compiled-function (&optional arg info params
execution even if an existing\nresult cached in the buffer would otherwise have been returnec
babel-get-src-block-info'.\n\nOptionally supply a value for PARAMS which will be merged with
org-babel-execute-src-block(nil)
(cond ((eq type 'headline) (cond ((memq (and (boundp 'org-goto-map) org-goto-
map) (current-active-maps)) (org-goto-ret)) ((and (fboundp 'toc-org-insert-
toc) (member "TOC" (org-get-tags))) (toc-org-insert-toc) (message "Updating table of content
safe (org-get-tags))) (org-force-cycle-archived)) ((or (org-element-property :todo-
type context) (org-element-property :scheduled context)) (org-todo (if (eq (org-
element-property :todo-type context) 'done) (or (car ...) 'todo) 'done)))) (org-
update-checkbox-count) (org-update-parent-todo-statistics) (if (and (fboundp 'toc-
org-insert-toc) (member "TOC" (org-get-tags))) (progn (toc-org-insert-toc) (message "Updati
before-first-heading-p) (line-beginning-position) (save-excursion (org-back-
to-heading) (point)))) (end (if (org-before-first-heading-p) (line-end-position) (save-
excursion (org-end-of-subtree) (point)))) (overlays (condition-case nil (progn (overlays-
in beg end)) (error nil))) (latex-overlays (cl-find-if #'(lambda ... ...) overlays)) (image-
overlays (cl-find-if #'(lambda ... ...) overlays))) (+org--toggle-inline-
images-in-subtree beg end) (if (or image-overlays latex-overlays) (org-clear-
latex-preview beg end) (org--latex-preview-region beg end)))) ((eq type 'clock) (org-
clock-update-time-maybe)) ((eq type 'footnote-reference) (org-footnote-goto-
definition (org-element-property :label context))) ((eq type 'footnote-definition) (org-
footnote-goto-previous-reference (org-element-property :label context))) ((memq type '(time
follow-timestamp-link)) ((memq type '(table-row table)) (if (org-at-TBLFM-
p) (org-table-calc-current-TBLFM) (condition-case nil (progn (save-excursion (goto-
char (org-element-property :contents-begin context)) (org-call-with-arg 'org-
table-recalculate (or arg t)))) (error nil)))) ((eq type 'table-cell) (org-
table-blank-field) (org-table-recalculate arg) (if (and (string-empty-p (string-
trim (org-table-get-field))) (and (boundp 'evil-local-mode) evil-local-mode)) (progn (evil-
change-state 'insert)))) ((eq type 'babel-call) (org-babel-lob-execute-maybe)) ((eq type 's
cookie) (save-excursion (org-update-statistics-cookies arg))) ((memq type '(inline-
src-block src-block)) (org-babel-execute-src-block arg)) ((memq type '(latex-
environment latex-fragment)) (org-latex-preview arg)) ((eq type 'link) (let* ((lineage (org-
element-lineage context '(link) t)) (path (org-element-property :path lineage))) (if (or (ec
element-property :type lineage) "img") (and path (image-type-from-file-name path))) (+org-
-toggle-inline-images-in-subtree (org-element-property :begin lineage) (org-
element-property :end lineage)) (org-open-at-point arg)))) ((org-element-
property :checkbox (org-element-lineage context '(item) t)) (let ((match (and (org-
at-item-checkbox-p) (match-string 1)))) (org-toggle-checkbox (if (equal match "[ ]") '(16))
in-regexp org-ts-regexp-both nil t) (org-in-regexp org-tsr-regexp-both nil t) (org-
in-regexp org-link-any-re nil t)) (call-interactively #'org-open-at-point) (+org-
-toggle-inline-images-in-subtree (org-element-property :begin context) (org-
element-property :end context)))))
(let* ((context (org-element-context)) (type (org-element-type context))) (while (and conte

```
through subscript superscript))) (setq context (org-element-property :parent context) type (
element-type context))) (cond ((eq type 'headline) (cond ((memq (and (boundp ...) org-
goto-map) (current-active-maps)) (org-goto-ret)) ((and (fboundp 'toc-org-
insert-toc) (member "TOC" (org-get-tags))) (toc-org-insert-toc) (message "Updating table of
safe (org-get-tags))) (org-force-cycle-archived)) ((or (org-element-property :todo-
type context) (org-element-property :scheduled context)) (org-todo (if (eq ... ...) (or ...
update-checkbox-count) (org-update-parent-todo-statistics) (if (and (fboundp 'toc-
org-insert-toc) (member "TOC" (org-get-tags))) (progn (toc-org-insert-toc) (message "Updati
before-first-heading-p) (line-beginning-position) (save-excursion ... ...))) (end (if (org-
before-first-heading-p) (line-end-position) (save-excursion ... ...))) (overlays (condition
case nil (progn ...) (error nil))) (latex-overlays (cl-find-if #'... overlays)) (image-
overlays (cl-find-if #'... overlays))) (+org--toggle-inline-images-in-subtree beg end) (if
overlays latex-overlays) (org-clear-latex-preview beg end) (org--latex-preview-
region beg end)))) ((eq type 'clock) (org-clock-update-time-maybe)) ((eq type 'footnote-
reference) (org-footnote-goto-definition (org-element-property :label context))) ((eq type
definition) (org-footnote-goto-previous-reference (org-element-property :label context)))
follow-timestamp-link)) ((memq type '(table-row table)) (if (org-at-TBLFM-
p) (org-table-calc-current-TBLFM) (condition-case nil (progn (save-excursion (goto-
char ...) (org-call-with-arg ... ...))) (error nil)))) ((eq type 'table-
cell) (org-table-blank-field) (org-table-recalculate arg) (if (and (string-
empty-p (string-trim (org-table-get-field))) (and (boundp 'evil-local-mode) evil-
local-mode)) (progn (evil-change-state 'insert)))) ((eq type 'babel-call) (org-
babel-lob-execute-maybe)) ((eq type 'statistics-cookie) (save-excursion (org-
update-statistics-cookies arg))) ((memq type '(inline-src-block src-block)) (org-
babel-execute-src-block arg)) ((memq type '(latex-environment latex-fragment)) (org-
latex-preview arg)) ((eq type 'link) (let* ((lineage (org-element-lineage context '... t)) (
element-property :path lineage))) (if (or (equal (org-element-property :type lineage) "img"
type-from-file-name path))) (+org--toggle-inline-images-in-subtree (org-element-
property :begin lineage) (org-element-property :end lineage)) (org-open-at-
point arg)))) ((org-element-property :checkbox (org-element-lineage context '(item) t)) (le
at-item-checkbox-p) (match-string 1)))) (org-toggle-checkbox (if (equal match "[ ]") '(16))
in-regexp org-ts-regexp-both nil t) (org-in-regexp org-tsr-regexp-both nil t) (org-
in-regexp org-link-any-re nil t)) (call-interactively #'org-open-at-point) (+org-
-toggle-inline-images-in-subtree (org-element-property :begin context) (org-
element-property :end context))))))
```

Printing the value of the `client` variable from inside `edebug` on `jupyter-kernel-info` yields

```
;; client  ;;pp-eval-last-sexp
#s(jupyter-org-client
   (#<finalizer>)
     jupyter--clients "idle" 1 #s(hash-table  size  65  test  equal  rehash-
size 1.5 rehash-threshold 0.8125 data
                                            ())
   nil #s(jupyter-channel-ioloop-comm
        (#s(hash-table size 1 test eql weakness value rehash-size 1.5 rehash-
threshold 0.8125 data
                        (t #0)))
         #s(jupyter-zmq-channel-ioloop
            (#<finalizer>)
            #<process zmq> nil
```

```
((send
  ((channel jupyter-channel)
   msg-type msg msg-id)
  ((list 'sent
         (oref channel type)
         (jupyter-send channel msg-type msg msg-id))))
 (stop-channel
  (type)
  ((let
       ((channel
         (object-assoc type :type jupyter-channel-ioloop-channels)))
     (when
         (and channel
              (jupyter-channel-alive-p channel))
       (jupyter-stop-channel channel))
     (list 'stop-channel type))))
 (start-channel
  ((channel jupyter-channel)
   endpoint)
  ((when
       (jupyter-channel-alive-p channel)
     (jupyter-stop-channel channel))
   (oset channel endpoint endpoint)
   (let
       ((identity
         (jupyter-session-id jupyter-channel-ioloop-session)))
     (jupyter-start-channel channel :identity identity))
   (list 'start-channel
         (oref channel type)))))
((setq jupyter-channel-ioloop-session
       (jupyter-session :id "38bcac68-f74f-4bd2-b1e7-
998df7c14c4f" :key "a1369b21-aa7ff7834dd1f3fa5f7108e7"))
 (require 'jupyter-channel-ioloop)
 (require 'jupyter-zmq-channel-ioloop)
    (push 'jupyter-zmq-channel-ioloop--recv-messages jupyter-
ioloop-post-hook)
 (cl-loop for channel in
          '(:shell :stdin :iopub)
          unless
     (object-assoc channel :type jupyter-channel-ioloop-channels)
          do
          (push
          (jupyter-zmq-channel :session jupyter-channel-ioloop-
session :type channel)
          jupyter-channel-ioloop-channels)))
 ((mapc #'jupyter-stop-channel jupyter-channel-ioloop-channels))
#s(jupyter-hb-channel :hb #s(jupyter-session
          (:shell_port 60543 :iopub_port 52071 :stdin_port 46591 :control_
aa7ff7834dd1f3fa5f7108e7" :transport "tcp" :signature_scheme "hmac-sha256" :kernel_name "ju
1.5")
                                    "38bcac68-f74f-4bd2-b1e7-
```

18

```
998df7c14c4f" "a1369b21-aa7ff7834dd1f3fa5f7108e7")
                                        "tcp://127.0.0.1:49243" #<user-
ptr ptr=0x6000002f88a0 finalizer=0x10e782ed0> 10 ignore t t)
          jupyter-zmq-channel-ioloop #s(jupyter-session
                          (:shell_port 60543 :iopub_port 52071 :stdin_port 46591 :control_
aa7ff7834dd1f3fa5f7108e7" :transport "tcp" :signature_scheme "hmac-sha256" :kernel_name "ju
1.5")
                                            "38bcac68-f74f-4bd2-b1e7-
998df7c14c4f" "a1369b21-aa7ff7834dd1f3fa5f7108e7")
          (:stdin #s(jupyter-proxy-channel "tcp://127.0.0.1:46591" t)
                :shell #s(jupyter-proxy-channel "tcp://127.0.0.1:60543" t)
              :iopub #s(jupyter-proxy-channel "tcp://127.0.0.1:52071" t)))
    #s(jupyter-session
      (:shell_port 60543 :iopub_port 52071 :stdin_port 46591 :control_port 37599 :hb_port 4924
aa7ff7834dd1f3fa5f7108e7" :transport "tcp" :signature_scheme "hmac-sha256" :kernel_name "ju
1.5")
      "38bcac68-f74f-4bd2-b1e7-998df7c14c4f" "a1369b21-aa7ff7834dd1f3fa5f7108e7")
    #s(hash-table size 65 test equal rehash-size 1.5 rehash-threshold 0.8125 data
                  ())
    #s(jupyter-kernel-process-manager
        (#<finalizer>)
        jupyter--kernel-managers #s(jupyter-command-kernel
                                  (#<finalizer>)
                                  ("julia-1.5" "/ssh:notebooks-vm.us-central1-
f.quarere|docker:klt-notebooks-vm-cjme:/home/jovyan/.local/share/jupyter/kernels/julia-
1.5" :argv
                                  ["/usr/bin/julia" "-i" "--startup-file=yes" "--
color=yes" "--project=@." "/home/jovyan/.julia/packages/IJulia/tOM8L/src/kernel.jl" "{conr
                        :env nil :display_name "Julia 1.5.1" :language "julia" :interrupt_m
                                #s(jupyter-session
                              (:shell_port 60543 :iopub_port 52071 :stdin_port 46591 :control_po
aa7ff7834dd1f3fa5f7108e7" :transport "tcp" :signature_scheme "hmac-sha256" :kernel_name "ju
1.5")
                                              "38bcac68-f74f-4bd2-b1e7-
998df7c14c4f" "a1369b21-aa7ff7834dd1f3fa5f7108e7")
                              #<process jupyter-kernel-julia-1.5>)
      #s(jupyter-zmq-channel :control #s(jupyter-session
                              (:shell_port 60543 :iopub_port 52071 :stdin_port 46591 :control
aa7ff7834dd1f3fa5f7108e7" :transport "tcp" :signature_scheme "hmac-sha256" :kernel_name "ju
1.5")
                                              "38bcac68-f74f-4bd2-b1e7-
998df7c14c4f" "a1369b21-aa7ff7834dd1f3fa5f7108e7")
                                        "tcp://127.0.0.1:37599" #<user-
ptr ptr=0x6000002f8900 finalizer=0x10e782ed0>))
    #<buffer  *jupyter-kernel-client*> nil "null" nil nil nil)
```

When I check all of the existing kernel files, I find one whose kernel ID is different (14238987-f1b4-4049-982a-94012ddb7087)from what is contained in the client variable (38bcac68-f74f-4bd2-b1e7-998df7c14c4f), but whose key and various ports are all correct.

```
/docker:klt-notebooks-vm-cjme:/home/jovyan/ #$ cat .local/share/jupyter/runtime/kernel-
14238987-f1b4-4049-982a-94012ddb7087.json
```

```
{
  "shell_port": 60543,
  "iopub_port": 52071,
  "stdin_port": 46591,
  "control_port": 37599,
  "hb_port": 49243,
  "ip": "127.0.0.1",
  "key": "a1369b21-aa7ff7834dd1f3fa5f7108e7",
  "transport": "tcp",
  "signature_scheme": "hmac-sha256",
  "kernel_name": "julia-1.5"
}
```

This suggests that the root of the problem is that the kernel ID is not being captured accurately. There is no kernel with an ID equivalent to the one that appears in the `client` variable, so it is not clear to me where the value that appears in the client variable is coming from. I don't know if there is any condition in which jupyter changes the kernel ID and leaves all other parameters the same.

The issue appears to arise in `jupyter-make-client` (see L141 in jupyter-kernel-manager.el) when calling `make-instance class` for class `jupyter-org-client` which derives from `jupyter-repl-client` which derives from `jupyter-kernel-client`. The `session` is defined as an element of `jupyter-kernel-client` at L215 of jupyter-client.el.

5   Which model construction process works as a whole?

We are working in an applied way to build models, starting with data and using existing tools and methods, but without any strong guarantee that we will find the most effective methods right away. So, with these experiments we are investigating the process of "model construction" generally understood. One example is building computational structures from natural language and technical texts.

5.1   Information extraction from SO Q&A items

We are attempting to extract triples from textual Q&A by using a Neural Machine Translation approach.

5.1.1   **BACKBURNER** Refinining OpenIE approach   DEYAN

5.2   Knowledge graph

Once we have a model of knowledge from Q&A items, e.g., in the form of triples. we will want to be able to do something with this material. One way in which it may be useful is in combination with an existing knowledge graph. For example, we can look at material from Concept Net. We may also have to make some of our own Concept Net-like graphs.

5.2.1   Practical work

We can already take some practical steps here, along the lines of the earlier papers "Modelling the way mathematics is actually done" and "Towards mathematical AI via a model of the content and process of mathematical question and answer dialogues".

5.2.2   **STARTED** Analyse a small sample of examples from s.o.   JOE

6 Underlying foundation

We believe that category-theoretic foundations will help us make progress across different representations of code, process, model building, and so on.

6.1 Category theoretic glue

We want to develop enough theory that we can use it to frame our experiments. We are trying to do this in a computationally meaningful way.

6.1.1 Feature: Understand comma categories as a potential "backend"   RAY:ZANS

6.2 Probabilistic programming for scientific modelling

Probabilistic programming is useful within both scientific modelling, and, potentially, as part of a program synthesis toolkit.

6.2.1 Feature: relationship between probabilistic programming and categories   ZANS:CAMERON

7 POTENTIAL PRODUCTS

Synthesis of some of our *projects* could lead to marketable *products*.

7.1 Agent model

One of our central intentions is to instantiate our work in an agent model of Q&A and programming. This is based on Alan Turing's suggestion that computers could talk with each other to sharpen their wits.

7.2 Recommender System

We could consume various analyses of Stack Exchange data to make recommendations.

7.2.1 Possible implementation strategy: build on a version of GPT fine-tuned on SO Q&A tasks

Could we set up a simple version of GPT trained on Stack Overflow data, just to get it working? Then think about how to get a learning loop set up to improve the results...

1. Ideas

    - Could this at least help a human navigate the questions on Stack Exchange?
    - Rather than just answering the question, generate the answer and use that to guide search (by combining generation with document similarity)
    - Use a distance to set up a margin of tolerance

2. Precedents

    - Stack Roboflow creates ersatz Q&A using `AWD_LTSM`. Surely we can do better?
    - In Google Books, they use crappy OCR which is good enough for search, but you wouldn't want to read the output. For search, they use something like rewrite distance, finding something 'within 5 errors'.

3. Analogue

    In parsing, it's not just edit distance but has to involve the grammar

4. Case against going too deep:

- Code generation is hard

5. Case against worrying about that:

   - Worry instead about applications like generating learning packets
     - E.g., learn everything there is to know about `git` from Stack Overflow in a nicely organised way.
     - E.g., compare the Schuam's Outline series: could we reassemble open source clones of Schuam's Outlines by retrieving contents from Math.Stack Exchange?

6. Application of the model: Display SO with similarity graph E.g., use generated answers to help identify 'similarity'.

7. Related work

   - https://github.com/stared/tag-graph-map-of-stackexchange/wiki presents a nice-looking map of the relationship between tags.

7.2.2 Feature: Initial import of SO for training TIM

7.3 Visual Interfaces

7.3.1 Graphical flow for programs

Can we model more general program flow in a similar fashion to Monocl?

7.3.2 Limitations

The idea of graphical programming languages is linked with the Deutsch limit (named for noted programmer L Peter Deutsch, not physicist David Deutsch FRS, though perhaps he could come into play later):

> *The problem with visual programming is that you can't have more than 50 visual primitives on the screen at the same time.*

7.3.3 Automatically create visual interfaces

Here's an idea: assuming we have enough text mining pixie dust (on corpora of linux man pages, and stack overflow questions/forum posts about linux commands), it might be possible to do:

```
user:~$ make-gui-for ls --output ls.py
```

7.3.4 Feature: Build infra for generating and displaying graphs.

E.g., we can generate graphs based on code flow.

```
(defun triangle (n)
  (if (equal n 0) 0
    (+ n (triangle (- n 1)))))
```

This would then be related to the visual code walk through feature described below.

Ray is working on a visual code walk through. This should be seen as another interface to the same basic underlying information, sort of like how Org Roam is the main interface to the data served by Org Roam Server.

1. General evaluation strategy for these demos:

   - *'Would anyone want to use this?'*
   - E.g., in the case of Emacs "learn X in Y" demo.
   - If there is interest, work up to covering the HtDP book

2. Related work

   MAUDE framework.   You describe your programming language using rewrite rules in K. They define tools to auto-derive rules in K.

   Program slicing 'Galois connection on the traces'. This allows you to find where bugs appeared. People tend to look in the most recent. Imagine a call-graph of all the variables, so it gives you a minimum trace, showing where your bug can be found.

7.4   Data course

There's a new book available from the group affiliated with STAN. It doesn't go very far, but it has tons of examples. They have data sets about all sorts of stuff. So the idea would be to take, e.g., the notebook on linear regression, and go through...

7.4.1   Idea

Start with a method, then go through lots of examples. Make this consistent with the way we would teach HtDP.

"Here's a data set, here's a method that would make sense to apply."

7.4.2   A quandry

Note that hand-coding of a curriculum vs making a general framework that anyone can contribute to (e.g., to make their own curricula) are pretty different things. We will sort out this ambiguity later.

7.4.3   Sources

There are tons of great data sets, but the issue would be digging into the details of some of them. The real issue is coordinating. We want to start with e.g., intro to linear regression, then hierarchical linear regression, and working up to things like Lotka-Voltera model.

- Datopian

7.4.4   How to build up to this?

- E.g., setting up the pre-requisites of the platform

- Setting up a tutorial on model building in a certain domain, get 10 people in the specialised tutorial, how is it received

- This would start building up the group of people

  - Using someone else's platform would be different from using our own platform
  - Which of these is the focus? (Good question but let's have one or two sprints beforehand to see where things are going.)

7.4.5 Assumptions

- Keep platform open source, assume people would want to use

7.4.6 Comments

- Platform is quite a general word, but in a way we are trying to make something easier

- The platform is just an interface to a piece of technology we build. The core is really on the backend.

- So the focus should be on the backend not on the javascript bits.

- Maybe leverage more existing technologies for the platform, where building it basically means installing it.

- Nextjournal: this looks good because they have UX designers to polish things

- Cloud-based Emacs: Would allow you to back your instantiation as if Emacs is your operating system, 500GB instance on Google Cloud

7.4.7 Status

- Cameron has code to set up a multicluster platform available off the shelf that we can start with

- Ray has been doing similar things for personal use, though if this helps write biology papers.

- What if our user interface was Emacs?

    - Different keybindings; developers like Emacs or Vi...
    - Org Bable exists & we can refer to this for now

7.4.8 Reference

- Michael Betancourt: Towards a principled bayesian workflow

7.5 Paperspace DO NJ etc. Collaboratory

This would be a potential user-facing product in which we could deploy various curricula, share various tools for interacting with scientific/computational models, and build a "knowledge hub" of people who could do scientific work.

8 BUSINESS DEVELOPMENT

8.1 Relationship to purpose

Understanding how the business activities relate to the purpose? We might do things that appear unrelated what we say at Why not what to serve customer needs in the mean time. However, if we do, we should either come up with some reasoning about how this helps us address the purpose, or revise our statement of purpose to reflect the current reality. This presumably isn't hard to do, e.g., we could say "once we have a successful business we will pour $x\%$ into research," but in any case we should clarify this.

8.2 Roughly B2C

- Launch some version of the Emacs Hyper Notebook as a cloud service. (Build it first and test it first.)

- Visual Interfaces: Develop a user interface on top of more advanced data analysis tools. (The focus is on the infrastructure that allows you to convert a graph into a neural network or whatever.)

- Data course (training format): Recruit people to take our course for a fee.

- Paperspace DO NJ etc. Collaboratory (Edtech SaaS): People would build their own courses/projects on our software and pay for licensing.

- Teach arbitrary coding (Edtech SaaS): People would use our tutoring system to improve their programming abilities.

8.3 B2B

- Agent model (software as a service format): We can run our agent model to generate new code or other insights. People can pay for compute plus a premium for quality.

- Probabilistic programming for scientific computing (Consulting format): going around and creating customers by talking to businesses, saying "Using proabablistic programming — or other technologies — we can optimize this, this, this, and this, saving you this much money."

  - Many companies hardly use any AI, let alone deep learning. If you can hustle and sell things, this can work.
  - However, we don't want to sell AI snake oil, so if we are going to do consulting it should be around topics that we're actually experts on. For example, plausibly, we could talk about modelling *documents* and *workflows*.

8.4 Different kinds of users

If we want to build a business, we should focus on who our target users actually are, and what problems we can solve for them. Typically we would build the business in a customer-centric way. So, for example, are the users/customers:

- Advanced STAN users, or,

- People who don't know how to do data analysis but who can make graphs.

  Broad categories of users are surveyed in the Downstream.

8.5 Related work

- Be wary of competing with things like Roam, though some level of competition is intrinsic in business.

- "Roam scratches my itches for document and graph aware note taking pretty well."

9 RESEARCH OUTPUTS

We would like to publish some papers, though as Deyan points out we should only do this when we have high-quality results:

> Deyan: *Every paper that is published for the sake of an academic's publication record, rather than for its scientific merit, is potent fuel for science denialism. The short-term shortcuts for a personal career, when compounded, cause long-term harm to the scientific endeavor.*

So, what can we do without shortcuts?

9.1 Advances in tutoring systems for programming

This would be a survey paper that would inform our efforts to Teach arbitrary coding. Follow references, start with 'AI and tutoring'.

1. (2014) "An adaptation algorithm for an intelligent natural language tutoring system"

2. (2008) "A novel approach for constructing conversational agents using sentence similarity measures"

9.2 Advances in knowledge mining from technical documents

This would be a survey paper that would inform our efforts on *Information extraction from SO Q&A items and the Knowledge graph approach. Note that if we can find survey papers that others have done, that's pretty much just as useful, and saves us a bunch of time.

9.2.1 **STARTED** Reading "Machine Knowledge" paper   DEYAN

9.3 An ABM of the computer programming domain

This would be a paper writing up our agent model work.

The paper could also correspond to a "whitepaper" that talks about how we are able to "mine" computer programs automatically. This would contribute to a long-term business in automated programming (and potentially other kinds of automation work).

10 Bottom

By the time we get to this point, we will have established some impressive research outputs, a potentially profitable business, and a teaching/upskilling platform for technical and scientific topics.

Figure 1: Network view

11 Downstream

What do our potential users look like?

11.1 Consulting clients

We discussed the idea of doing consulting for clients who are interested in using scientific models.

• Play through again as a consulting client

11.2 Scientific software developers

We imagine some software developers consuming "tutorial" content we produce, and improving their skills and abilities as a result.

• Play through again as a scientific software developer

11.3 Automated tutoring system users

We imagine some students using AI software we develop. In some cases they could be "students". In other cases, they could already be professional developers.

• Play through again as an automated tutoring system user

11.4   Programmers

We imagine any programmer having some use for our tools. "B2D" (Business to Developer) is an emerging category of enterprise where we can do interesting things.

- Play through again as a programmer

12   Organisational infrastructure

This section is mildly-technical appendix. It looks at our organisational infrastructure itself, including simple things like the technologies we use for communication, and more involved things like "how we communicate" more broadly. (This is a good candidate for splitting off into its own separate wiki, if for no other reason than that it takes up a lot of space in the generated PDF.)

12.1   Schedule and activities

Presently we are meeting 20 minutes a day at 4PM UK time, 11AM Eastern, on Discord for a "coffee chat". Previously we tried to maintain a schedule of longer meetings (UK evenings):

- Monday: Seminar

- Wednesday: Workshop

- Friday: Studio

That seemed to be too many meetings. Whatever we do about regularly scheduled meetings, we might want to look at how to best pursue of *topics of mutual interest such as:

- Readings on rewriting rules and production systems, and higher-dimensional graph-like things

- Business development around open source, knowledge management, etc.

- Reviewing the value add of Wiki ways of thinking and working, which we have a pretty broad range of experience with

- R&D around 'lenses' in ACT: structure for bi-directional transformations, to enable changes in a projection

So far, this Roadmap has gathered information on some of the topics that have been discussed, but not all of the things that we could see ourselves working on together.
As another activity we may want to get scheduled one or more sessions focused on business stuff.

12.2   Project orientation

Some of this will be different depending on whether we think of this as a "business", or as "a business of some specific nature": primarily centring on "who does this business do business with?"

- Status - where is the project right now?

    - Right now *this overall project* is in a "project development" mode.

    - What are the (multiple) *success indicators* or *proof points* or *failure indicators* for each of the projects? (E.g., going to the casino with $20, you might quit when you get below $10, you might leave when you get above $50.) E.g., need of customers for X, our credibility in X?

    - For the various sub-projects: one relevant thing is "how long is it before thing is likely to make money?" (AKA, "Cross-over.") Or "what else is needed for this to make money?"

- In particular: maybe take a couple months to see how things are going with a given sub-project? This gives evidence of what we can produce when we work together. We might then ask, who else would care to pay for this?
- We have listed 4 active projects (`https://miro.com/app/board/o9J_kmPNvaQ=/`); maybe the blog is another one.

- Roles and Responsibilities - *who is handling the standard project roles, and what are they responsible for doing?*

  - Each individual sub-project is likely to have different requirements (e.g., some may need 2 people, some will need 1, etc.)
  - If there's more than one person involved it becomes a parallel architecture

- Goals - *What will this project achieve?*

  - "If I do something valuable, the money will come later."
  - Some of them we might be willing to take the risk of investing time and energy based on whether it looks directly useful to us.
  - Some, like a course, we may need the information about whether it's likely to be taught.
  - Some could become a paper or the building block of a business: these can be small demo projects.
  - Alternatively, in a consulting mode, our role becomes understanding customer goals and helping rationalise work to fulfil them.

- Resource Requirements - *What (people, money, things) are needed to accomplish this project? Where do they come from?*

  - We each individually need some money, but it's not totally clear that the *company* needs some money.
  - If we wanted to replace any one of us with an employee, then we'd have to have some funding source.
  - If the number of person-hours for the goal is quite high, then it's unlikely for the goal to be achieved without funding.
  - E.g., what would we need to be able to do consulting?

- People - *Who are the people working on this project? Who can I ask for more information? How can I best get in touch with them?*

  - If we were to be doing consulting, then it becomes about serving specific customer needs.

- Approach - *What is the overall strategy for accomplishing this project?*

  - Whatever we choose (e.g., consulting vs product development) we should choose it based on some data and analysis.
  - Wherever we are now, the question is what's needed to move ahead.

- Workplan and Timeline - What are the specific tasks needed to accomplish our goals? When might they happen? Who / what / when (in agile, we specify two).

  - Joe needs some job soon!
  - To do consulting we'd need to figure out customer need and credibility
  - To make progress on the AI directions we need some version of all the things up and running!

- Communication Norms - how have the project participants agreed to stay in touch? what, where and how often are regular meetings? Special ceremonies?

  - In 2 months we'll have 2 more months of experience. So we could then assess things.

  - In advance of that, we might start to understand the expections about how we would gather the data.

  - It should be pretty much fun, and if it's not we're kind of doing it wrong?

  - On an ongoing basis we should be able to check whether what we're doing is effectively addressing the goals we have

- Sponsor - *the person who requires the output of the project and has allocated the resources for it (aka Customer in agile)*

  - So far we're all sponsoring our own work on sweat equity

  - While also trying to be helpful & respectful to each other

  - EF was the sponsor at one time

  - Joe provided chips and dip but the event was strictly BYOB... as long as we're here we'll make the best out of. Polka time!

- Project Manager - the person responsible for the drumbeat and tempo of the project, and for its administrative details, including good project management hygiene

- Lead - the person responsible to the Sponsor for making sure the project is accomplished and to the Team for making sure they are able to accomplish the project

  - Ray: project to build bridges between participants (e.g., systems bio, category theory, stats); this is related to the "transdisciplinary design" course

  - Joe: I'm less technically sophisticated

- Team - people working on the project

  - Everyone will have some constraints (like need $40K per year if it takes more than 20 hours per week)

12.2.1   Project Management Hygiene

- set SMART goals (Specific, Measurable, Achievable, Relevant and Time-based)

- understand tasks required to accomplish goals, then set realistic timeline

- create project plan in wiki

- regular, frequent check-ins to iterate plan (goal, priorities, etc.) if necessary

- after-action reviews at the end of project, including reflection/writeup of positives and deltas

- experienced, well-oiled teams requires less strict project management hygiene

- new, less-organized, or heterogenous teams require more attention to careful project management hygiene

12.2.2   **TODO** Make a list of actual topics of interest   ALL

If we were just doing "content production" we might think of a list of chapters to write, or podcasts to produce. However, maybe those ways of thinking and working don't apply comfortably here.

12.3   Technology

Does <https://github.com/orgs/exp2exp/projects/1> conflict, replace, or serve a different function compared with Org mode agenda items?

12.4   Discord server

We set up a Discord server that we're using for our meetings. This invite link should not expire: <https://discord.gg/pArjt4p>

(We also have a Zulip server set up, but currently we're using it less.)

12.5   OBS recordings

We talked about creating asyncronous recordings (screencasts, audio). We also talked about possibly putting the audio recordings into a threaded voice mail forum, but that's a somewhat different application.

12.6   Code sharing platform

For now we have a Github organisation (<https://github.com/exp2exp>), as well as a separate repo that contains these Org Roam notes, among other things. This could potentially be improved or upgraded in various ways.

12.6.1   Comments

- Nextjournal is interesting

- It's like a Jupyter notebook

- It's like Org Bable so you can run code in any language within the same environment

- If I need to add a bash cell to a Julia notebook, it adds a kernel as needed at the run time

- If I install a bunch of libraries, and save the current environment in a docker container, you can import it

- It doesn't yet have an easy way to make an app?

12.6.2   What if you had a browser based version of Org Bable?

- You could have your notebook, backed by the ability to use Emacs

12.6.3   Examples

- Setting up a data science experiment

- Wadler et al. course in Agda in NextJournal

- But you can't easily treat this as 'Org Roam' (no bi-directional things)

12.6.4   Next evolution

We need a basic code sharing platform to get to work. The next evolution might look like what we've been calling the "Emacs Hyper Notebook"? However, some contributors are not interested in using Emacs for everything. And we can't assume that users would be interested in it either!

12.7 Wiki

The public facing version of these notes is available on a simple web interface, created by firn: https://exp2exp.github.io/. This mirrors the contents of our Org Roam directory. Editing is explained below.

We can also view the contents of Org Roam in a linear form as PDF document... or view the currently active tasks using Org Agenda. In the future we may want to have several different "upstream" locations, based on several different small-scale wikis, all feeding into this one location. That's not hard to set up. Contents can also be browsed in a graphical form either with the built in `org-roam-graph` functionality, or by installing Org Roam Server and running `org-roam-server-mode`.

We can potentially improve on all of this further, bulding something like Metacademy. For now, we describe how to use this simple Org Roam based wiki.

12.7.1 Access

Obtain the sources by cloning the repo at https://github.com/exp2exp/exp2exp.github.io.

```
git clone git@github.com:exp2exp/exp2exp.github.io.git
```

(See below for an alternative.)

12.7.2 Mob branch on repo.or.cz

We're mirroring the repo to an environment that allows anonymous commits (without need for further permissioning). If want to contribute anonymously, info on that is here: https://bit.ly/2EQRHEF

You can review commits to the mob branch here: https://repo.or.cz/arxana.git/shortlog/refs/heads/mob

12.7.3 Setup

Install Org Roam if needed (`M-x package-install RET org-roam RET`).

Subsequently, add this to your Emacs configuration:

```
(require 'org-roam)
(setq org-roam-directory (concat "/home/"
                                 (getenv "USER")
                                 "/exp2exp/"))
(setq org-roam-completion-system 'helm)
(define-key org-roam-mode-map (kbd "C-c n l") #'org-roam)
(define-key org-roam-mode-map (kbd "C-c n f") #'org-roam-find-file)
(define-key org-roam-mode-map (kbd "C-c n b") #'org-roam-switch-to-buffer)
(define-key org-roam-mode-map (kbd "C-c n g") #'org-roam-graph)
(define-key org-mode-map (kbd "C-c n i") #'org-roam-insert)
(org-roam-mode +1)
```

12.7.4 Bonus feature: org-roam-checkout

If you regularly use your own separate Org Roam setup, you can use this simple context switcher to move between the two. Keep track of the various separate Org Roam installations with `org-roam-library` and then switch between them interactively with `org-roam-checkout`.

```
(defvar org-roam-library `(,(concat "/home/" (getenv "USER") "/exp2exp/")
                           ,(concat "/home/" (getenv "USER") "/org-roam/")))

(defun org-roam-checkout ()
  (interactive)
```

```
  (let ((ctx org-roam-directory))
    (if (eq (length org-roam-library) 1)
        ;; Still go ahead and set the variable in this case!
        (progn (setq org-roam-directory (car org-roam-library))
            (message "You only have one choice for org-roam-directory defined."))
      (let ((lib (completing-read "Choose a volume: " org-roam-library)))
        (when lib
          (setq org-roam-directory lib))))
    ;; assuming the user changes context, let's also prompt them
    ;; to choose a new file in that context
    (when (not (eq ctx org-roam-directory))
      (org-roam-find-file))))
```

12.7.5 Interaction

Use the `C-c n f` keyboard command to add new disconnected nodes to the graph, or use `C-c n i` to create a page and insert a wiki-style link, like `[[New Page]]`. Follow links with `C-c C-o`. Display the graph structure with `C-c n g`. It may be necessary to run `M-x org-roam-db-build-cache` to get the graph to match reality. Add and commit new or modified files with git, along with `org-roam.db`, and push them to the repo.

12.7.6 Tags

Some of the nodes have `#+roam_tags` set:

code	meaning
HL	High level
CDN	Can do now
LRD	Longer R&D cycle
HD	Has dependencies
RR	Research Review
RO	Research Output
OTS	Off the shelf
SH	Stakeholder
AN	Annex

Some of the files also have a `#+CATEGORY` set.

12.7.7 Pairing

For syncronized browsing and editing with lockstep.el:

```
ssh pair@178.79.174.58
PW: <ASK JOE FOR THE PASSWORD>
emacsclient -a '' -t
M-x lockstep
```

To open up a real-time collaboration (with multiple cursors), use crdt.el, first to serve the buffer:

```
M-x crdt-serve-buffer
```

And then, from your client, to connect:

```
M-x crdt-connect
```

12.7.8 Linearizing

To turn this map into something we can reliably use, let's try to linearize it.

To downsample from Org Roam (save as `~/bin/roam2org.sh` and make it executable):

```bash
#! /bin/bash

emacs --batch -l ~/bin/downsample-org-roam.el --eval "(combine-files)" "$@"
```

Here are the working parts (save as `~/bin/downsample-org-roam.el`):

```lisp
(defun downsample ()
  "Process an Org Roam buffer for inclusion in a standard Org file.
Changes title to header, and increase indentation of existing headers.
Changes file links to internal links."
  (if (looking-at "^#\\+TITLE:")
      (replace-match "*"))
  (forward-line 1)
  (if (looking-at "^#\\+roam_tags:\\(.*\\)")
      (replace-match ":PROPERTIES:
 :tag:\\1
 :END:"))
  (while (re-search-forward "^\\*" nil t)
    (replace-match "**"))
  (goto-char (point-min))
 (while (re-search-forward "\\[\\[file:\\(([^]]*\\))\\]\\[\\(([^]]*\\))\\]\\]" nil t)
    (replace-match "[[*\\2][\\2]]"))
  (buffer-substring-no-properties (point-min) (point-max)))

(defun combine-org-roam-files (&rest args)
"Combine a list of files, specified as ARGs.
The files are to be found in `org-roam-directory'."
  (apply #'concat
         (mapcar (lambda (file)
                   (save-window-excursion
                     (find-file (concat org-roam-directory file))
                     (let ((contents (buffer-substring-no-properties (point-min)
                                                                     (point-max))))
                       (with-temp-buffer (insert contents)
                                         (goto-char (point-min))
                                         (downsample)))))
                 (or (car args) (nthcdr 5 command-line-args)))))
```

12.7.9 Backlog

Part of the idea with a backlog is to go from most-doable, starting with work in progress, to least-doable and potentially vague. Here, then, is one approximate linearization that may or may not meet that description!

Note, this is duplicated in the index file, probably for sanity we should pick one and automate the derived version from there!

```lisp
(defvar files-to-combine
'("20200810131435-hyperreal_enterprises.org"
"20200810132653-top.org"
```

```
    "20200905124558-why_not_what.org"
     "20200909195629-teach_arbitrary_coding.org"
     "20200810135851-how_to_design_programs_with_if.org"
    "20200905124405-construct_critique_improve_models_of_the_creative_process.org"
       "20200905125342-emacs_hyper_notebook.org"
       "emacs_jupyter_remote_debugging.org"
    "20200905125023-which_model_construction_process_works_as_a_whole.org"
     "20200905131027-information_extraction_from_so_q_a_items.org"
    "20200905131918-knowledge_graph.org"
    "20200905124432-underlying_foundation.org"
     "20200905125713-category_theoretic_glue.org"
     "20200905131656-probabilistic_programming_for_scientific_modelling.org"
    "20201003205523-potential_products.org"
     "20200905130423-agent_model.org"
     "20200817172825-recommender_system.org"
     "20200810135457-visual_interfaces.org"
     "20200814203551-data_course.org"
     "20200905132603-paperspace_do_nj_etc_collaboratory.org"
    "20200814210243-business_development.org"
    "20200905134325-research_outputs.org"
     "20200810135325-advances_in_tutoring_systems_for_programming.org"
     "20200810135403-advances_in_knowledge_mining_from_technical_documents.org"
     "20200905132334-an_abm_of_the_computer_programming_domain.org"
    "20200906003704-bottom.org"
     "20201003164408-downstream.org"
     "20201003165500-consulting_clients.org"
     "20201003170312-open_source_developers.org"
     "20201003170333-tutoring_students.org"
     "20201003171011-programmers.org"
    "20200810135126-organisational_infrastructure.org"
     "20200810135619-discord_server.org"
     "20200811185435-obs_recordings.org"
     "20200814193042-code_sharing_platform.org"
     "20200912223428-wiki.org"
     "20201003164100-forum.org"
     "20200814195259-blog.org"
    "sfi/sfi.org"
     "sfi/gather_data_via_stack_exchange_apis.org"
     "sfi/argumentation_theoretic_analysis.org"
     "sfi/process_model_analysis.org"
     "sfi/ml_nlp_bootcamp.org"
     "sfi/initial_ml_baseline_e_g_match_q_a.org"
     "sfi/hierarchical_ml_for_content_extraction.org"
     "sfi/active_inference_bootcamp.org"
     "sfi/agent_modelling_and_sandbox_setup.org"
     "sfi/curate_koans_and_develop_solver.org"
     "sfi/study_with_crowdsourced_exercises.org"
     "sfi/study_with_agent_written_questions.org"
     "sfi/publication_ijcai.org"
    )
    "An ordered list of files to combine in our export.
```

```
This is where the order of presentation in the downstream org file
and derived PDF is defined.")
```

To combine the files, run:

```
(combine-org-roam-files files-to-combine)
```

To get the indicative nesting (shown by spaces above) to be replicated at the org level, run the following at the top of the exported compilation:

```
(defun indent-org-roam-export ()
  "Utility function to increase indention for selected trees."
  (org-map-entries (lambda ()
                    ;; don't demote the top level items and their sub-items
                    (let ((tag (org-entry-get nil "tag")))
                      (if (and tag (string= (car (split-string tag)) "HL"))
                          (progn (org-end-of-subtree)
                                 (setq org-map-continue-from (point)))
                        (org-do-demote))))
                   nil 'file))
```

Lastly, to rebuild the PDF, all of this can be done with one swift action.

```
(defun rebuild-org-roam-pdf ()
  "Build an org file and PDF compiling `files-to-combine'."
  (interactive)
  (save-excursion (find-file (concat org-roam-directory
                                     "/manual/combined.org"))
    (goto-char (point-min))
    (search-forward "# IMPORT")
    (let ((beg (point)))
      (delete-region (point) (point-max))
      (insert "\n" (combine-org-roam-files files-to-combine))
      (goto-char beg)
      (indent-org-roam-export)
      (org-latex-export-to-pdf))))
```

12.7.10   Publishing to the web

Publishing with Firn is simple:

```
firn build
```

Then commit and push.

12.7.11   Reviewing progress

Something like the following should be all that's get a high-level overview of progress on active tasks, sourcing information directly from the Org Roam files. Add the following to your emacs initialisation script (e.g., `~/.emacs`), evaluate it, and then run `C-c r` to load up the fun. This may not be the perfect presentation yet but it gives an idea.

```
(setq org-todo-keywords
      '((sequence "TODO" "STARTED" "BLOCKED" "BACKBURNER" "FROZEN"
                  "|" "DONE" "DEFERRED" "WONTFIX")))

(setq org-agenda-sorting-strategy '((todo todo-state-down category-down)))

(setq org-agenda-files '("~/exp2exp/"))

(defun org-scrum-board ()
  (interactive)
  (org-todo-list "TODO|STARTED|BLOCKED|BACKBURNER|FROZEN|DONE|DEFERRED|WONTFIX"))

(global-set-key (kbd "C-c r") 'org-scrum-board)
```

This view can then be further filtered by regexp (e.g., your name) by pressing =.

12.7.12 **DONE** Package downsamping code separately   JOE

12.7.13 **WONTFIX** Update the repo instructions to reference this file   JOE

12.8 Forum

We talked about using Wikum as a forum, because we liked the idea of a workflow based on summarising discussions. There's now a demo instance set up that we can use, here:

http://wikum.org/visualization_flags?id=590&owner=holtzermann17

12.8.1 Could we incorporate the ideas directly in Org or Org Roam?

Perhaps we could incorporate some Wikum ideas right into the wiki here. The idea would be to treat the top paragraph on each page as a summary, and then add discussion threads below. We'd want some system of tags that indicated whether the summary was validated or now. (Note the the original WikiWikiWeb did not have separate talk pages! I don't know if they practiced robust summarisation, either.)

1. REMARK   JOE This is an
   "inline task," via `(require 'org-inlinetask)`. There doesn't seem to be support for nested or
   threaded tasks, but maybe we would have use for non-threaded forum discussions at the end of any
   page in the Wiki. Incidentally, for those curious, the formatting of the LaTeX export is controlled by
   `org-latex-format-inlinetask-function`.

2. END

12.8.2 **TODO** What might our summarisation workflow look like?   ALL

Since we're pretty actively updating our Discord and pretty happy using it, maybe people who are working on Active Projects would be willing to summarise on the wiki, say, weekly? And contribute to a monthly group blog post?

12.9 Blog

This is a public window on our experiments, available at https://exp2exp.com.

Presently, we're still figuring out what the work flow and contents of the blog will look like. The kinds of people to whom we wish to appear credible are described in Downstream, and presumably whatever we put online should match what we think they will want to know.

Zans: *If I implemented as I read things, it would be a pretty interesting blog. There could be a huge market of people interested in following this, this would give a pool of people who know who we are. This is a nice goal b/c it doesn't focus on the product... but it's a deliverable, made up of smaller deliverables, and a concrete benefit.*

13 Joe's Santa Fe Institute proposal (Redux)

This is some content trying to build a linear plan. From within the Hyperreal Enterprises repo this should connect to Top. Inside the Roadmap document, this now works as an appendix. It uses the first person (I=Joe).

13.1 Gather data via Stack Exchange APIs

This is just a matter of downloading the data in bulk. We've also seen that some of the data is available in a bulk download. Tim did some work to load it into a machine learning framework.

13.2 Argumentation-theoretic analysis

I've started to look at some of the Q&A items to see whether there's anything there we can extract.

This isn't yet the same as an argumentation-theoretic analysis, but I can imagine returning to this Q&A stuff through the lens of the earlier papers, and see what we could do to argument-mine them. That could lead to the "FARM 2.0" paper that Ray and I have often discussed.

13.3 Process model analysis

This is about making a model of the process of programming, per Monocl, which I think we've been starting to understand a bit better through HtDP recently.

13.4 ML/NLP bootcamp

I got some insights from both Tim and Deyan about what is directly possible to do with existing tools. We didn't yet progress beyond some initial "failed" experiments and partial work. However, in principle we could go back to it and try to do Q&A matching. That's a low-level task.

13.5 Initial ML baseline, e.g., match Q-A

This remains a pretty straightforwardly "doable" task if we were to get things loaded up. After taking a stab at things within the the HEL sessions, I could presumably take that as a curriculum to be able to get this running after several months (with adequate help from Tim and Deyan). However, it's less obviously useful... if we can't do the "hierarchical ML" next step.

13.6 Hierarchical ML for content extraction

I guess part of this would be to do first Coarse Discourse style detections, then move from that to more specific models. During the HEL sessions I don't think we've made any directly tangible progress on this goal. I did find references to *CROKAGE*, which I should follow up on. Still, maybe there are some alternative ways to make progress here that should be thought about.

13.7 Active Inference bootcamp

Cameron talked a bit about active inference, and his Bayesian stats background is relevant to making sense of what's going on in here. Lots of other papers are available, including some by Beren Millidge if we wanted to start making models of agents.

We could also/alternatively think about these models in terms of Tangled Program Graphs or some other reinforcement learning paradigm. E.g., Zans and I talked about making an upvoting bot. If we could moreover learn the actual features that go into good questions and answers, then those could be used on the generative side too.

13.8 Agent modelling and sandbox setup

This seems to be the main outcome of the Active Inference bootcamp.

Could we set up a simpler agent learning paradigm, and progressively run it in long epochs, in which the agents learn more complicated skills each time? E.g., learn how to upvote, learn how to retrieve relevant concepts, learn how to structure them into sentences and paragraphs?

Maybe it's worth looking around at some of the other Stack Exchange sites to get inspired, or have a think about what domains the Q&A approach might be especially suitable for. (E.g., are there any open source SQuAD-performant systems that would naturally lend themselves to agent modelling?)

Or, what about the question of turning Stack Exchange into a set of rewrite rules first, and basing the agents on those rules? If we think about things in terms of rewriting, then maybe each rewriting step could be viewed as a Q/A pair. So then, we could look at any open source rewrite system and try to give a score to rules based on how often they are used.

13.9 Curate koans and develop solver

The first part seems relatively straightforward: we already have questions available from sites like 4Clojure. But the solver hasn't really been that much in scope for any of the things we've been working on yet! However, Ray looked at some classic work on rule-based solutions to subtraction problems, and I'd say that something along these lines could be created for solving Clojure koans (each of which is mainly meant to teach a specific rule, anyway). I'm sure it would be helpful for me to get more practice, and this time, include a careful writeup of the 'rules' I follow.

13.10 Study with crowdsourced exercises

"Crowdsourced reading comprehension questions and simple exercises derived from existing Q&A would provide a further route to evaluation."

So the idea here would be to pay people to come up with some SQuAD-like questions and answers based on existing Stack Exchange problems. This would help build a catalogue of the kinds of things we're looking for inside of the Q&A items.

13.11 Study with agent-written questions

Can agents come up with questions to ask about a given question that break it down into more do-able components? Can we gain leverage by "Asking questions about Stack Overflow questions"?

What sorts of Q/A structures might go in to reading and comprehending a given question? Maybe, again, we could think in terms of rewrite rules. "Here is a portion of text that is unclear. Let's try to see what the author was actually trying to say."

13.12 Publication: IJCAI

If some of the foregoing ideas "worked" then we would presumably have some nice things to write up for a publication. In fact, a bunch of the other ideas that were in my plan would probably be better saved for future work. Maybe things, like institutions, should be foregrounded. But, I can see how the plan I submitted to SFI probably came across as overly ambitious and unrealistic. Maybe the basic plan above would have to be spread out over 2 years.

On the other hand, some of these ideas do look like fun exploratory kinds of topics, which should maybe be thought about in some brainstorming sessions, just to keep things fresh. So, for now, these could become slips of paper that we pull out of a jar at a party to discuss?

13.12.1 Other possible future work

- M13 Institution modelling using IAD
- M14 Integrate themes from SFI collaborators
- M15 Develop infrastructure for contributors
- M16 Agents writing agents
- M17 Agents writing institutions
- M18 Organise first contest
- M19 Publication: Artificial Intelligence
- M20 Integrate with Github API
- M21 Integrate themes from SFI collaborators
- M22 Study in an online tutoring application
- M23 Publication: Science
- M24 Time off and plan Year 3