Project Information

- https://sites.cs.ucsb.edu/~tyang_class/293s22f/projects.html
- Develop a system prototype or algorithm implementation for search using a dataset or multiple datasets.
  - You may study algorithmic solutions or system performance related issues.
  - You may mix different systems and combine together through some simplified mechanism/assumptions
- Leverage open source code.
- Apply evaluation metrics to assess the success.
- Demonstrate the challenge of problem addressed, and leverage state-of-art technology
  - Recent technical paper(s) published in top venues SIGIR, WWW, WSDM, ACL, EMNLP.
  - Weakness study/improvement: better grade

Project Timelines

- Form a 2-person team and develop a project plan, find papers to study.
- Meet with me around Oct 19 to discuss about
  - The project topic
  - Paper selection
  - Dataset and what can be leveraged
  - Resource need
- Present the selected paper and your project briefly in late Nov.
- Final project report/short demo in Week 2 of Dec

Final Project Submission Requirement

- Demonstrate your project with me and submit the following project material by the end of the quarter:
  - Slides for the paper(s) presented.
  - A project report with at most 6 pages. The report needs to include
    - 1) Objectives+challenges.
    - 2) State-of-art techniques you have leveraged (The citation should include author names, title, where they publish, year).
    - 3) Key algorithms+techniques used with examples.
    - 4) Data set +metrics+evaluation results/your findings.
    - 5) Your efforts in the project and how the project is related to the class material learned.
  - Code and data sets used
  - Instructions for building and executing your demo so that your results are reproducible

Projects in the previous 293S course

- DeepTileBars: Visualizing Term Distribution of Neural Information Retrieval
- Deep Neural Networks for YouTube Recommendations
- Context Attentive Document Ranking and Query Suggestion
- Towards Better Text Understanding and Retrieval through Kernel Entity Salience Modeling
- A Comparison of Pretrained Language Models for Document Ranking
- Contextualized Embeddings for Document Ranking
- Examples will be posted in the website

You may use open source search engines

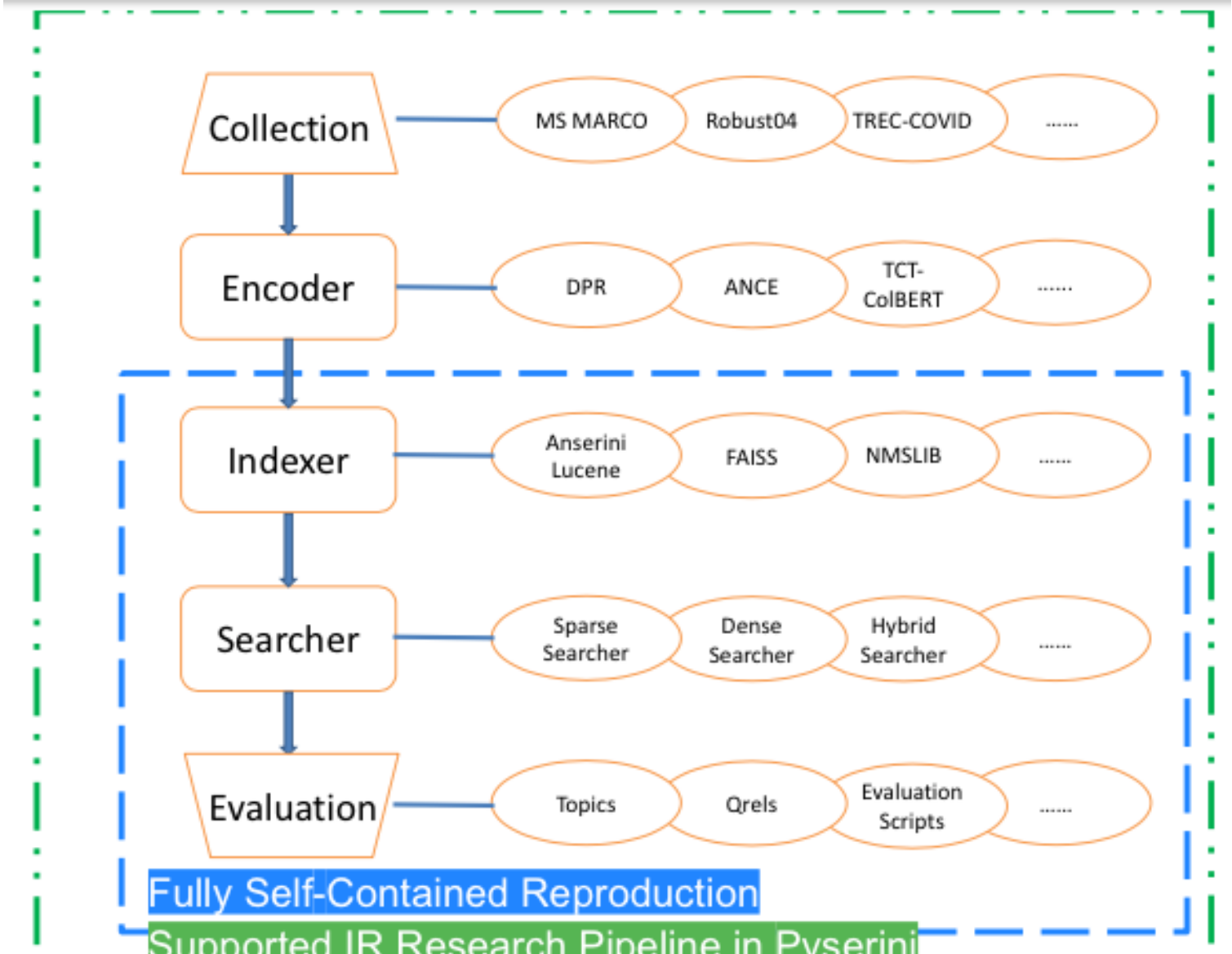
- **Lucene**
  - A full-text search library with core indexing and search services based on Java
- **Solr**
  - based on the Lucene Java search library with XML/HTTP APIs
  - caching, replication, and a web administration interface.
- **ElasticSearch:** based on Lucene
- **Lemur/Indri:** C++ search engine from CMU/U. Mass
- **Pyserini:** Python/Lucene based. U. Waterloo

Pyserini: https://github.com/castorini/pyserini/

- **Pyserini: An Easy-to-Use Python Toolkit to Support Replicable IR Research with Sparse and Dense Representations in SIGIR 2021 by** Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep and Rodrigo Nogueira. https://arxiv.org/abs/2102.10073

- Python based
- Open-source toolkits to evaluate retrieval models.
  - run as standalone IR tool or imported as library.
  - Mainly for the first-stage retrieval within a multi-stage ranking architecture .
  - Combines dense retrieval and sparse retrieval
- Provides access to Lucene data structures and system internals.
- Contains extensive regression tests that are run periodically to ensure reproducible result for several datasets

Pyserini Architecture



Pyserini: Examples

https://github.com/castorini/pyserini/blob/master/docs/experiments-robust04.md

Reproducing Robust04 Baselines

```
from pyserini.search import SimpleSearcher
searcher = SimpleSearcher.from_prebuilt_index('robust04')
hits = searcher.search('hubble space telescope')
# Print the first 3 hits:
for i in range(0, 2):
    print(f'{i+1:2} {hits[i].docid:15} {hits[i].score:.5f}')
```

```
1 LA071090-0047 16.85690
2 FT934-5418 16.75630
3 FT921-7107 16.68290
```

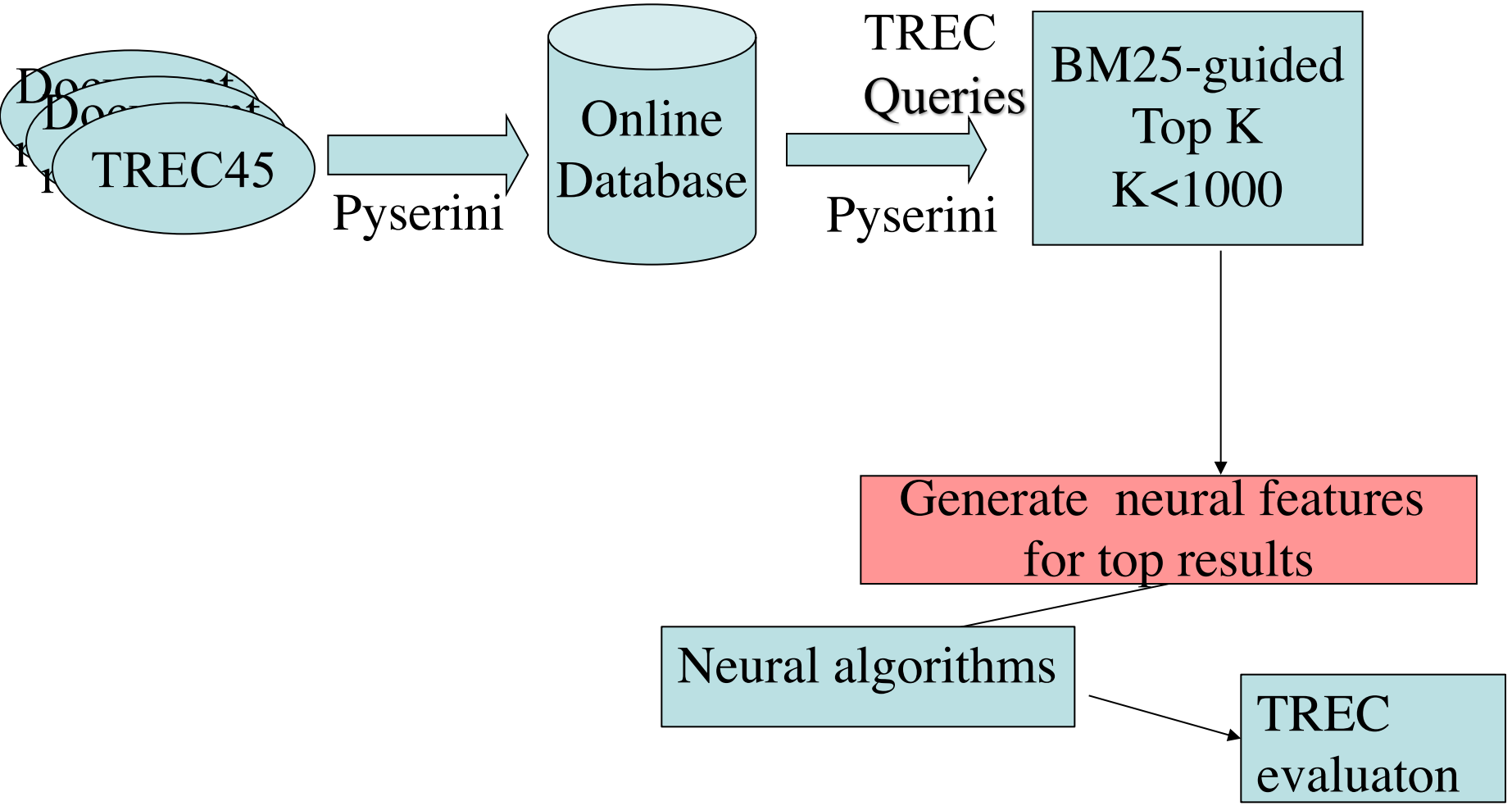
A batch retrieval run with a TREC evaluation tool:

```
$ python -m pyserini.search --topics robust04  --index robust04 --output run.robust04.txt --bm25
```

```
MAP  all 0.2531
P_30  all 0.3102
```

https://sites.cs.ucsb.edu/~tyang_class/293s23f/expanse293.html has info on how to run Pyserini on Expanse

Example Idea: Rerank top K results after retrieval using Pyserini

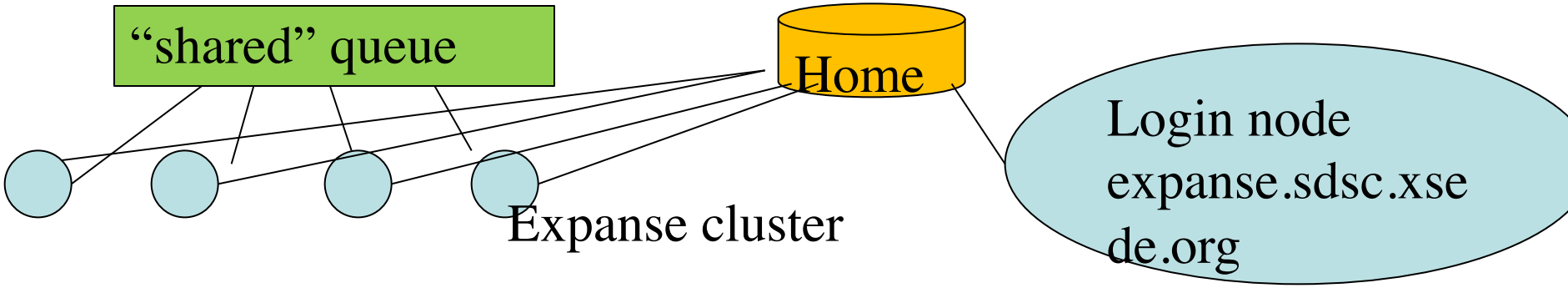


Computing Resource

- You can use any computing resource available to you, including your own laptops.
- **Google Colab: https://colab.research.google.com/**
  - write and execute Python in your browser, with free GPU.
- **CSIL machines with no GPU**
  - You may request for some extra disk space for a large dataset
- **Expanse cluster**
  - I have some CPU/GPU hours allocated for this course
  - www.cs.ucsb.edu/~tyang_class/294s23f/expanse293.html

How to Run a Job at Expanse Cluster

https://www.sdsc.edu/support/user_guides/expanse.html



- Expanse cluster managed by XSEDE has 728 standard CPU nodes, 54 GPU nodes
- ssh username@login.expanse.sdsc.edu
  - Login node can only be used for light activities
  - For example, can compile C code, but not Java code
- Sample example at Expanse under /home/tyang/293sample/
- Submit a job to run code at “shared” partition in the cluster
  - To submit a job that runs a hello binary
  - sbatch run-hello.sh

Job shell script that runs a sequential hello program at Expanse

```
#!/bin/bash
#SBATCH --job-name="hello"
#SBATCH --output="job_hello.%j.out"
#SBATCH --partition=shared
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --export=ALL
#SBATCH --account=csb175
#SBATCH -t 00:01:00

../hello
```

Job name you can observe when querying status

Job id in the submitted queue

Which cluster to run this program

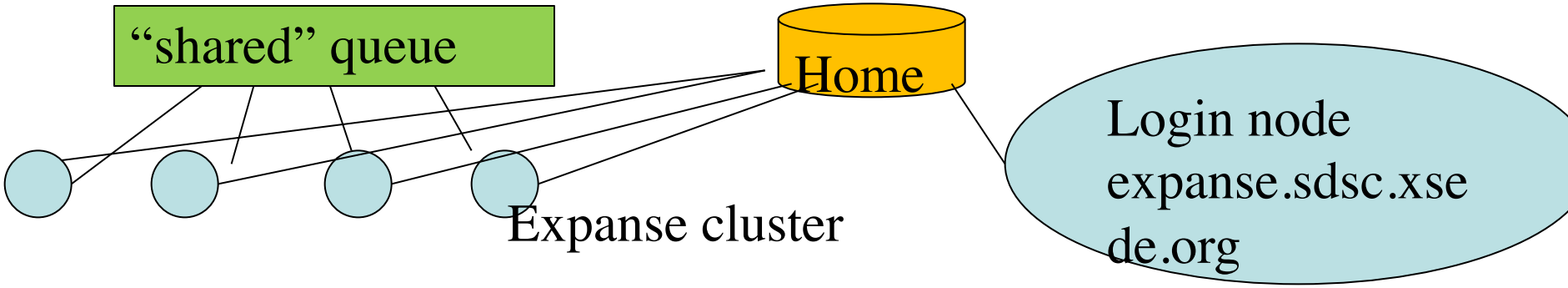
This job runs with 1 nodes, 1 core per node under CS140 account csb175

Set a time limit that this job runs at most 1 minute.

Run the hello program

```
main(void){
        printf("Hello\n");
}
```

How to get an account at Expanse



- Create an account at access-ci.org
- Fill your account name by a deadline in a spreadsheet I will distribute in Piazza
- I will add your account name as part of my class allocation
- After half a day or a day, you will see in the expanse cluster that you have a class-shared computing resource allocation.
- Then you can submit a job.

Other useful information regarding Expanse

command	notes
expanse-client user -p	Check balance
sbatch run-hello.sh	Submit job
squeue –u username	Check the job queues
sacct -u yryang	Check status of your jobs
scancel JobID	Cancel job
srun --pty --nodes=1 --ntasks-per-node=1 --cpus-per-task=10 -p debug -t 00:10:00 -A csb175 /bin/bash	Interactive mode
srun --pty --nodes=1 --ntasks-per-node=1 --cpus-per-task=10 -p gpu-debug --gpus=1 -t 00:10:00 -A csb175 /bin/bash	