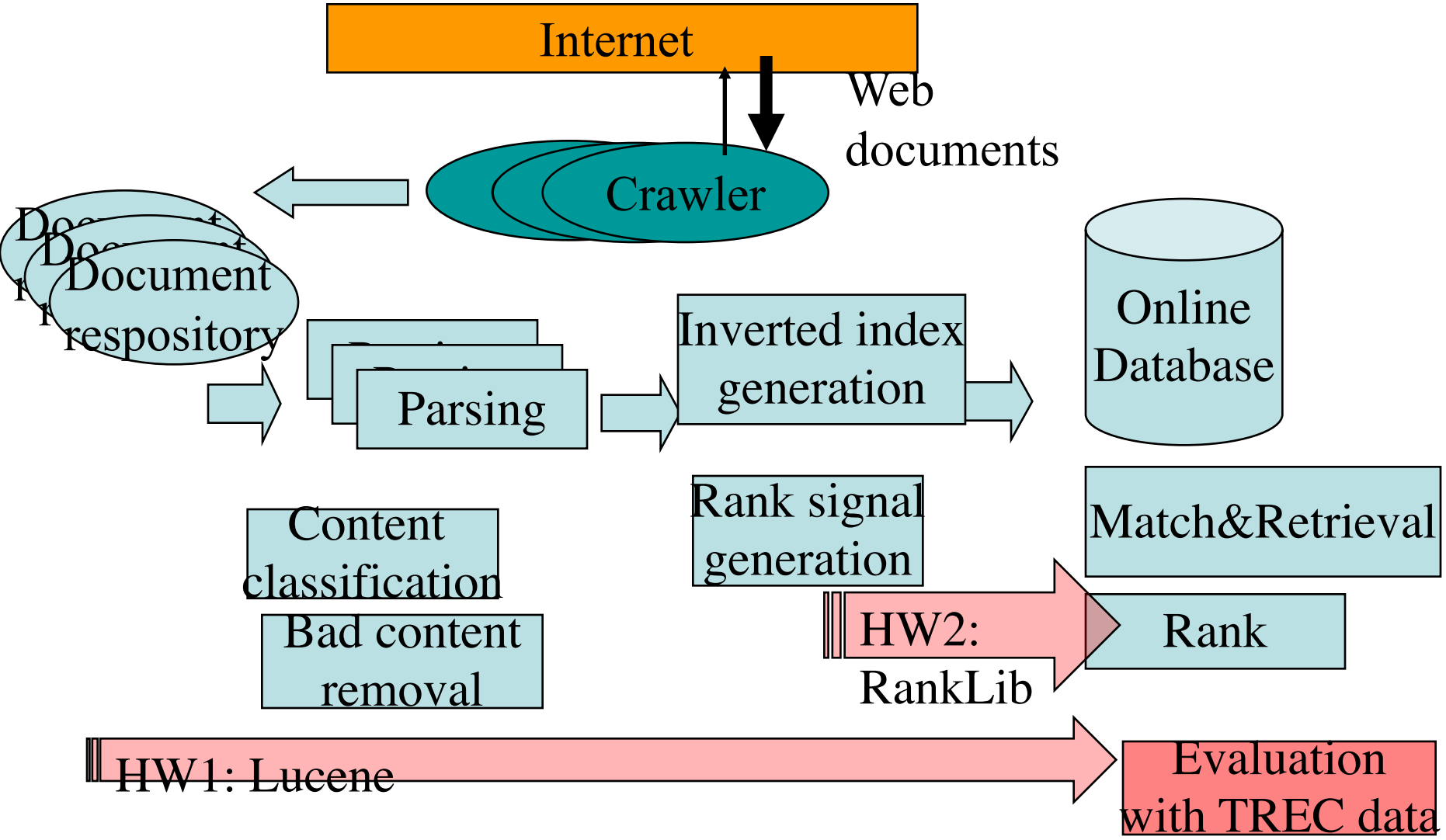
From HW1&2 to Project



Project Information

- https://sites.cs.ucsb.edu/~tyang_class/293s20f/projects.html
- Develop a system prototype or algorithm implementation for search or mining using a dataset or multiple datasets.
  - You may study algorithmic solutions for ranking and mining, 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 (e.g. recent technical paper(s) published in SIGIR, WWW, WSDM, KDD,ACL, EMNLP, SIGMOD/VLDB, etc or in another top venue).
- The CSIL machines can be used to build your project and there is a sandbox directory you can use a large amount of space

Timelines

- Form a 2-person team and develop a project plan, find papers to study.
- Meet with me in the first week of Nov to discuss about the project progress and paper selection. Through this discussion, we may select or assign a paper for presentation suitable for other students to learn on a related topic.
- Present a selected paper and your project briefly between mid Nov and earlier Dec

Timelines

- 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

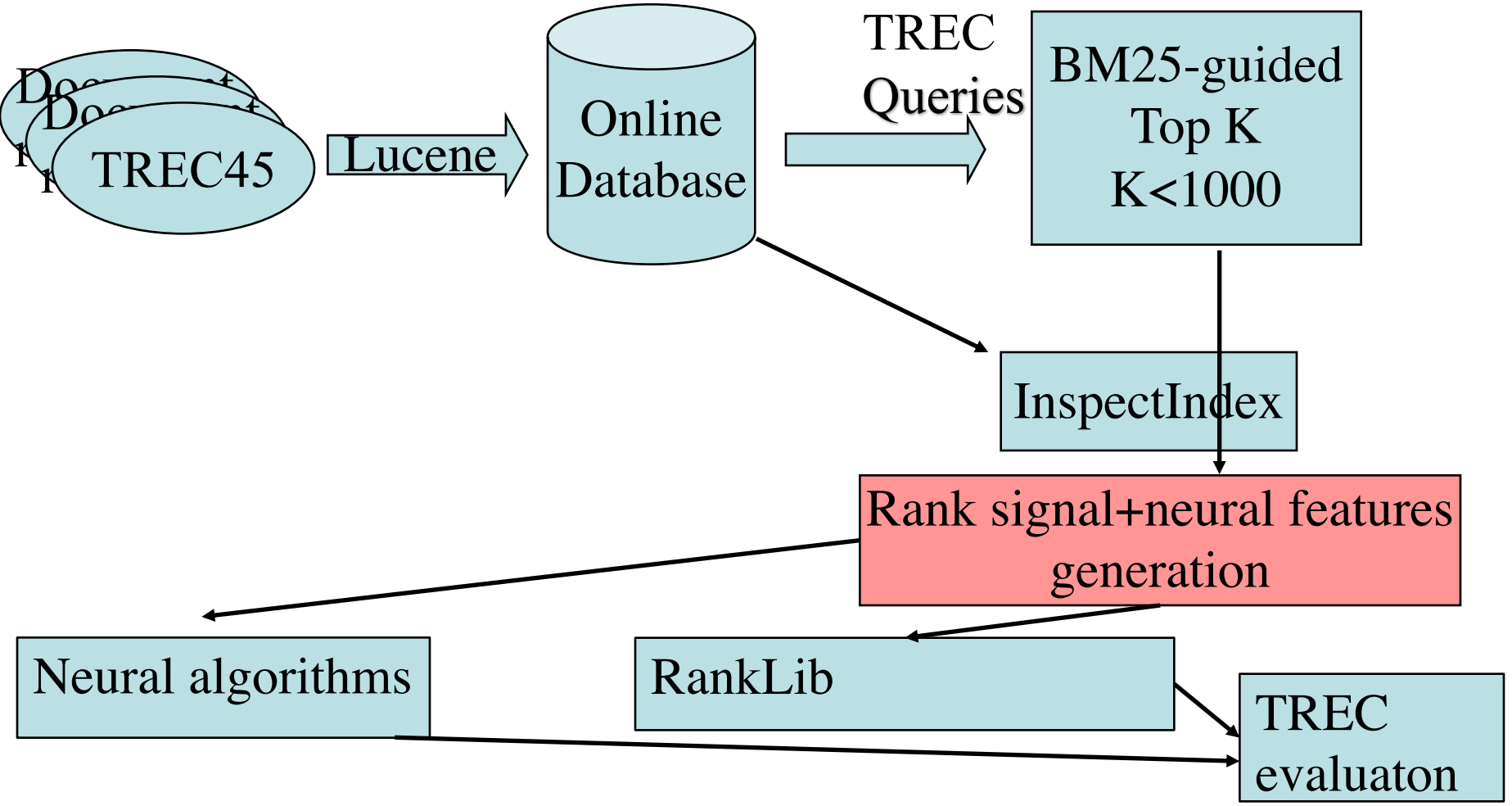
Projects in the previous 293S course

- Multi-stage Ranking with BERT
- 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
- Document Reranking based on HW1 with neural methods
- Pretrained Language Model for Document Ranking
- Contextualized Embeddings for Document Ranking

Some of Sample Project Ideas from https://sites.cs.ucsb.edu/~tyang_class/293s20f/projects.html

- **Check the above URL for more info**
- **Document Reranking based on HW1**
  - From HW1, you can derive a list of top-500 document IDs for each tested query and there are 150 queries.
  - You can gather the related documents for these queries from the trec45-processed.html, and build a set of new features such as neural text features or knowledge entity features for each document. Rerank top documents for each query.
  - You do not have to build an inverted index (which takes time to program). You can leverage HW1 Java code and simply retrieve the text of top documents and for each query, build necessary features for top documents saved in memory.

Example Project: Rerank top K results with advanced signals and ranking algorithms



Neural Indexing and Ranking Ideas

- **Different ways of indexing**
  - Sparse neural inverted index representation for documents (CIKM 2018)
    Use neural features to build inverted index and conduct TFIDF or BM25 ranking. The datasets used here also include TREC45
- Multi-stage ranking with BERT by Nogueira et al.
  The dataset and some code on how to replicate this work are available from here, using MS Macro-Passage-Ranking package.

Efficiency Improvement for Search

- **Handling multi-key word queries efficiently** on SSD
  - Wang et al., Evaluating List Intersection on SSDs for Parallel I/O Skipping VLDB 2020.

- **Fast boost tree algorithms**
  - http://ai.stanford.edu/~wzou/kdd_rapidscorer.pdf. KDD 2018 paper: RapidScorer: Fast Tree Ensemble Evaluation by Maximizing Compactness in Data Level Parallelization
  - QuickScorer https://github.com/hpclab/quickscorer
    - Quickscorer: a fast algorithm to rank documents with additive ensembles of regression trees SIGIR 2015
    - Source code is available.

Other Topics

- **Similar document clustering with word embeddings.**
  - From Word Embeddings To Document Distances by Kusner et al. ICML' 2015. The code and dataset Some recent work on how to Speeding up Word Mover's Distance and its variants via properties of distances between embeddings

- **Privacy-preserving top K search**
  - S. Ji, et al. Privacy-aware Ranking with Tree Ensembles on the Cloud . SIGIR 2018) Slides
    J. Shao, et al. Privacy-aware Document Ranking with Neural Signals. SIGIR 2019. Slides