Introduction to Legal IR

ESSIR 2023 – Legal IR Work Group

30 Aug. 2023

Sophia Althammer

Arian Askari

Amin Abolghasemi

Tobias Fink

Alaa El-Ebshihy

Agenda

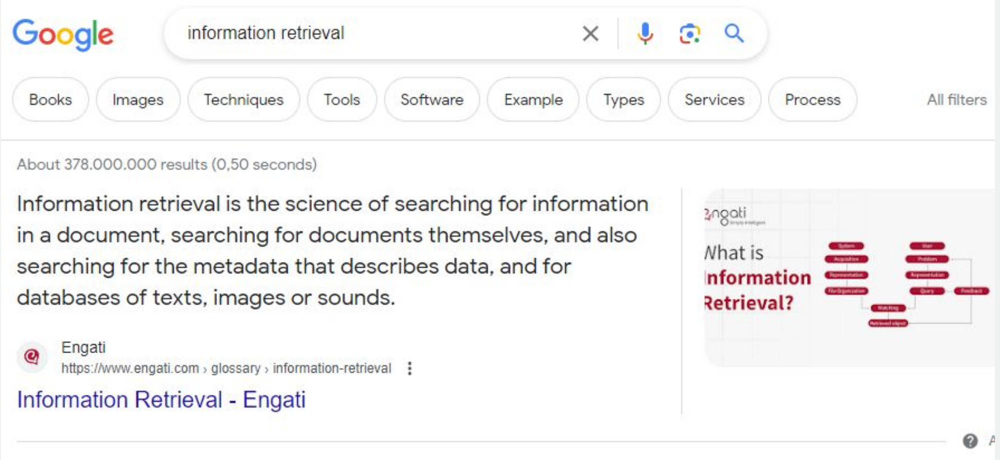
- Introduction
- Legal IR Tasks & challenges
- Evaluation campaigns, data sets & approaches
- Work group task

Agenda

- **Introduction**
- Legal IR Tasks & challenges
- Evaluation campaigns, data sets & approaches
- Work group task

Domain specific information retrieval

IR not only about web search



IR includes applications in other domains

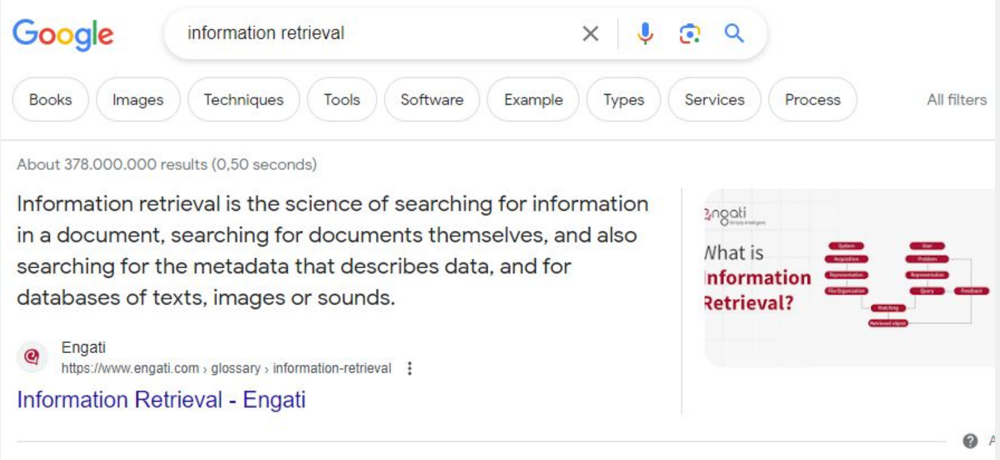
Medical

Legal

Scientific

Domain specific information retrieval

IR not only about web search



IR includes applications in other domains

Medical

Legal

Scientific

Agenda

- **Introduction**
- **Legal IR Tasks & challenges**
- Evaluation campaigns, data sets & approaches
- Work group task – COLIEE legal entailment

Legal Information Retrieval

"is the science of information retrieval applied to legal text, including legislation, case law, and scholarly works" – Wikipedia

- Differences to web search:
  - Users: Legals, paralegals
  - Queries: about legal cases
  - Search in: legal documents (e.g. laws or previous cases)

Different legal systems

Statue Law Vs. **Case Law**

Statue Law

- Statutes and legal regulatories are the primary information source
- Statute retrieval is important
- In European countries

Case Law

- Precedent cases are a primary source for legal evidence
- Prior case retrieval of high importance
- In Canada, US, Australia

Prior case retrieval in case law systems

- Task should lead to prior cases which should be noticed for solving the current case
- Information source is primary literature containing previous court decisions
- Desired output of the search is a list of prior cases, sorted by relevance or temporal aspects
- Precision-oriented task

eDiscovery

- eDiscovery in legal litigation with the requirement that documents be produced as critical evidence in litigation of a case
- Subject to rules of civil procedure and agreed upon processes, often involving review for privilege and relevance before data are turned over to the requesting party
- High recall demanded

Challenges

- Legal domain specific language

  Domain specific language models

    LegalBERT: trained on legal documents from CourtListener

- Long documents
  - Relevance on paragraph-level
  - Create summaries of cases

Agenda

- Introduction
- Legal IR Tasks & challenges
- Evaluation campaigns, data sets & approaches
- Work group task

Evaluation Campaigns and research datasets

- TREC Legal[1]

  eDiscovery for a production request

- COLIEE[2]

  Competition on Legal Information Extraction/Entailment for case law and statute law systems for Canadian and Japanese law systems

- FIRE AILA Track[3]

  Precedent & Statute retrieval for Indian case law system

1 TREC Legal Track, https://trec-legal.umiacs.umd.edu/
2 COLIEE Competition on Legal Information Extraction/Entailment https://sites.ualberta.ca/~rabelo/COLIEE2021/
3 FIRE AILA Track, https://sites.google.com/view/aila-2020/task-1-precedent-statute-retrieval

Neural IR approach

- Similar as in web search:

  1. First stage Retrieval
     - Boolean queries or
     - Dense retrieval approaches
  2. Transformer-based Neural re-ranking
     - Domain specific language models
     - Domain specific relevance annotations
     - Handling long documents with summaries or on paragraph-level by splitting up

Agenda

- Introduction
- Legal IR Tasks & challenges
- Evaluation campaigns, data sets & approaches
- Work group

FIRE AILA

Task 1B: Identifying relevant statutes

Identify the relevant statutes to query.

- Given:
  - query *Q* – short description of a legal situation
  - 197 statutes *S* – Sections of Acts from Indian law
- Output: the most relevant statutes from *S* to *Q*

Dataset:

- *Query_doc.txt* — 50 queries (Q)
- Object_statutes — contains the title and the textual description of 197 statutes
- *task1b_rel_judgements.txt* — relevance judgemnt in TREC format

Structure of today

FIRE AILA Task 1B: Identifying relevant statutes

- 9.30-10.00 Data examples and text pre-processing.
- Small Break
- 10.15-11.15 First Stage Retrieval: BM25
- Small Break
- 11.30-12.30 First Stage Retrieval: Splade
- Lunch Break
- 13.30-14.30 Neural Re-ranking and Evaluation: Finetuning BERT Cross encoders
- Small Break
- 15.00-16.30 Neural Re-ranking and Evaluation: Evaluating and comparing different crossencoders
- 16.30-17.30 (Optional) Few-shot prompting large language models for the task

Data examples

Task 1B: Identifying relevant statutes

- Task: **FIRE 2020 AILA -** Task 1B: Identifying relevant statutes
  - Given:
    - query Q – short description of a legal situation
    - example: „This appeal arises from the judgment of the learned Single Judge of High Court dated 6th June, 1988 whereby the learned Single Judge declined to quash the prosecution of the petitioner. The petitioner therein has been prosecuted for selling adulterated supari on the basis of a certificate issued by the Director of Central Food Laboratory showing that the article of Food purchased from the accused contained 2000 mgs/kg.....“

Bhattacharyaa, P., Mehtab, P., Ghoshc, K., Ghosha, S., Pald, A., Bhattacharyae, A., & Majumderf, P. (2020). Overview of the FIRE 2020 AILA Track: Artificial Intelligence for Legal Assistance.

Data examples

Task 1B: Identifying relevant statutes

- Task: **FIRE 2020 AILA -** Task 1B: Identifying relevant statutes
  - Given:
    - query *Q* – short description of a legal situation
    - 197 statutes *S* – Sections of Acts from Indian law
    - example of relevant statute: „Title: Penalty for giving or taking dowry
      Desc: 1 [(1)]If any person, after the commencement of this Act, gives or takes or abets the giving or taking of dowry, he shall be punishable 2 [with imprisonment for a term which shall not be less than 3 [five years, and and with fine which shall not be less than fifteen thousand rupees or the amount of the value of such dowry, whichever is more] "

Output: the most relevant statutes from *S* to *Q*

Bhattacharyaa, P., Mehtab, P., Ghoshc, K., Ghosha, S., Pald, A., Bhattacharyae, A., & Majumderf, P. (2020). Overview of the FIRE 2020 AILA Track: Artificial Intelligence for Legal Assistance.

Data preprocessing

Task 1B: Identifying relevant statutes

- Normally data cleaning is the most importatnt and crucial part, with research datasets some others have done this task (hopefully) for us, but thats not normal in production use cases
- For most neural IR libraries, standard data formats in tsv files are required:
  - Query and collection format are „query_id \t query_text \n“
  - Training triples are „query_text \t positive_text \t negative_text \n“
- For trec_eval (a standard evaluation toolkit) you need the relevance judgements in qrels format „query_id 0 doc_id relevance_grade“

Bhattacharyaa, P., Mehtab, P., Ghoshc, K., Ghosha, S., Pald, A., Bhattacharyae, A., & Majumderf, P. (2020). Overview of the FIRE 2020 AILA Track: Artificial Intelligence for Legal Assistance.

Creating the training triples

Task 1B: Identifying relevant statutes

- For sampling the training triples we take a relevant passage as positive text for a query
- Negative sampling can be done:
  - randomly
  - BM25 negatives
  - hard negatives
  - negatively labelled passages
  - ...
- Since we have negatively labelled passages in FIRE AILA we take those as negatives for the triples

Bhattacharyaa, P., Mehtab, P., Ghoshc, K., Ghosha, S., Pald, A., Bhattacharyae, A., & Majumderf, P. (2020). Overview of the FIRE 2020 AILA Track: Artificial Intelligence for Legal Assistance.

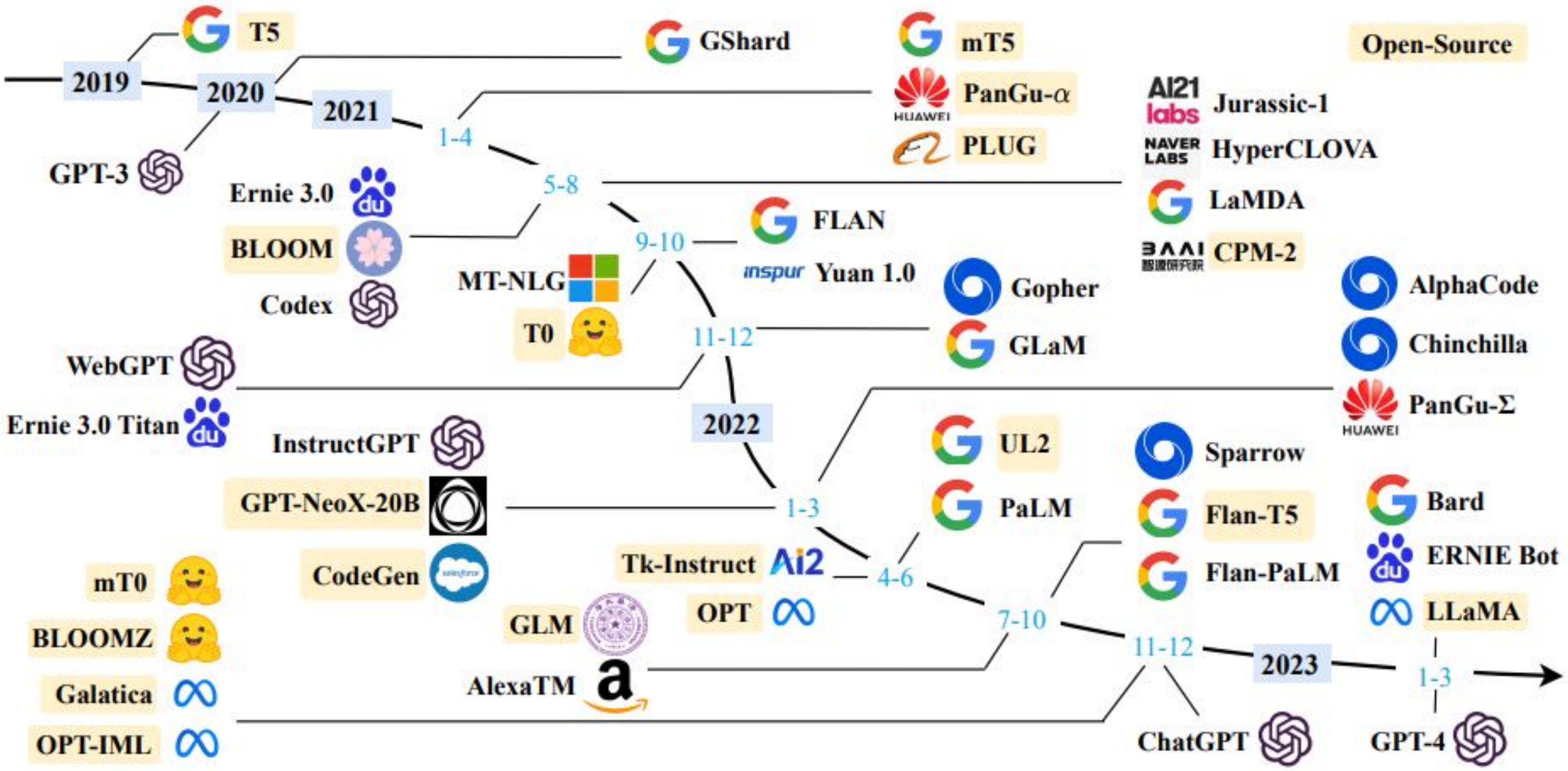
Sparse Retrieval

- **Objective:**
  - Implement code for searching statutes for given queries using Elasticsearch
  - Fill out missing parts in notebook, focusing on using basic Elasticsearch search methods
  - Experiment with Neural Network based sparse representations (SPLADE) and integrate them using the Elasticsearch API
- **Dataset:**
  - FIRE 2020 AILA Dataset: https://sites.google.com/view/aila-2020/dataset-evaluation-plan
  - Task1 Training Data includes queries, documents and relevance judgements for evaluation
  - Focus on Task1b: Statute Retrieval

  - **Decryption Passphrase:** ailasearch

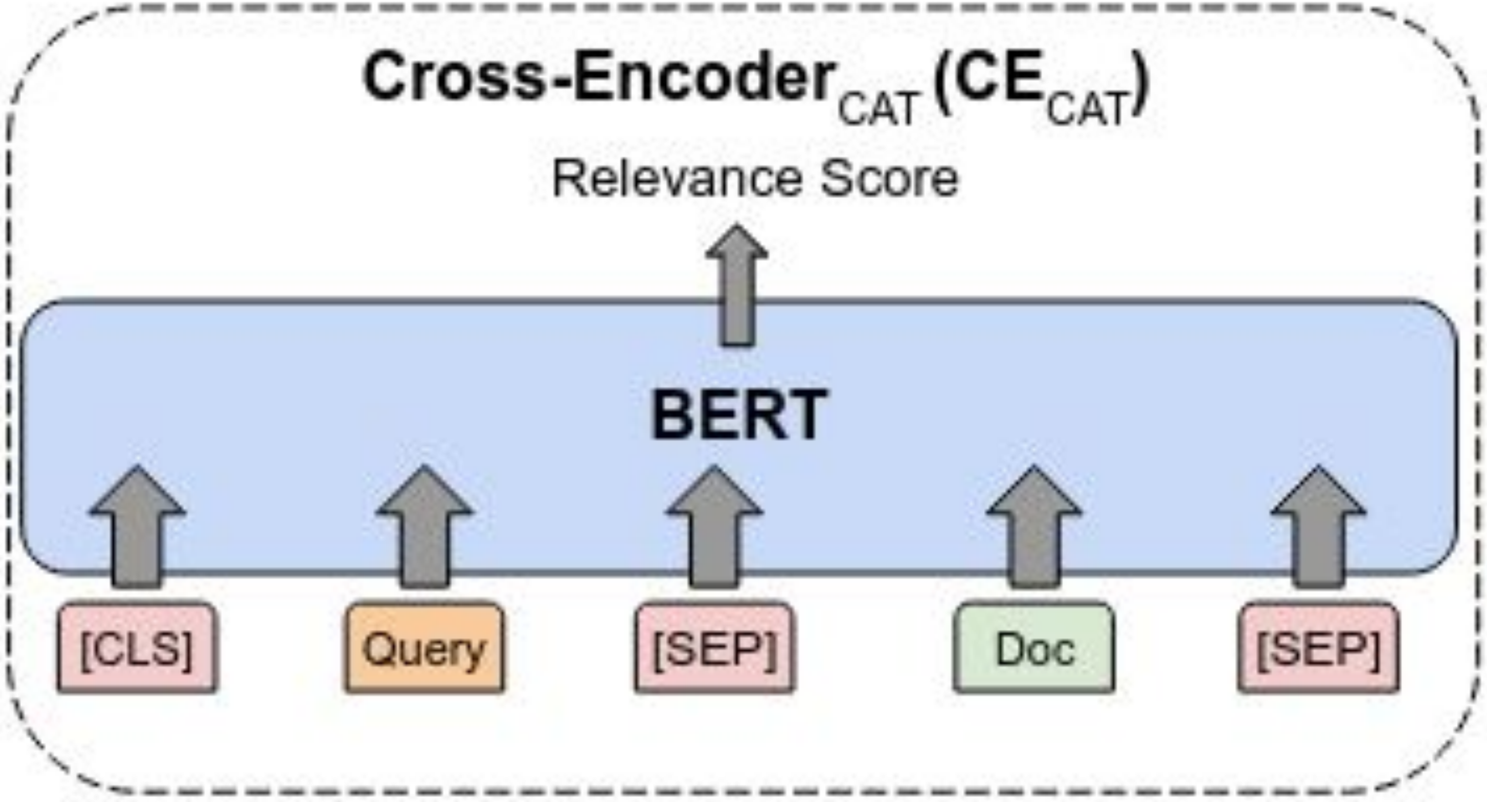
Afternoon session: LLMS and Cross-encoders for statute-reranking

- Open source large language models



Afternoon session: LLMS and Cross-encoders for statute-reranking

- Cross-encoder



Afternoon session: LLMS and Cross-encoders for statute-reranking

- What we will do
  - **Cross-encoders:**
    1. Training your own re-ranker
    2. Evaluating your own re-rankers compared to the first stage retriever (BM25)
       - **Try to guess:** What would be more effective?
         a. A reranker trained on one million web search queries? (start from a variation of BERT, called MiniLM)
         b. A reranker trained on 40 legal queries? (starting from legalbert)
    1. Analyzing the results and compare different models to each other
  - **Large language models:**
    1. Working with few-shot learning and Large language models in order to tackle re-rankin
       - Few shot learning = Training size = 2 instances
    2. **Designing your prompt and select a LLM**
    3. Assessing the effectiveness of the selected LLM on the task statute reranking
    4. Suggested LLMs (but you can explore by yourself):
       - T5-base (or small)
       - Flan-T5-base (or small)

Training Cross-encoders

- **First step:** Load **1_train_cross_encoder_essir_legalbert_assignment**.ipynb file on the Google colab (placed in afternoon_session folder)
- **Goal:** Realizing the implementation and completing tasks in order to train your own reranker
- **Note:** We will provide an already trained model on the full data so if you couldn't train you model completely. You will still be able to do evaluation.
- **Task 1:**
  - Completing part of evaluator class using Pytrec_eval library (Metrics to be considered: Map, P@1,5,10, Recall@10,100, and NDCG@10, 100)
- **Task 2:**
  - Truncating the content. Each query and document should be truncated to 254 words
- **Task 3 (optional if you found more time):**
  - Train cross-encoder with a small modification, inject the BM25 score into the middle of input, inspired by the following paper:
    - Askari, Arian, et al. "Injecting the BM25 Score as Text Improves BERT-Based Re-rankers." European Conference on Information Retrieval. Cham: Springer Nature Switzerland, 2023.
    - **We will help you for implementation if you started this task.**
- **Finally, solution:** https://colab.research.google.com/drive/nbtWUDSOO6XX9VboVk4HeMrZQ8aLeH0?usp=sharing

Evaluating Cross-encoders

- **First step:** Load **2_eval_cross_encoder_essir_assignemnt.ipynb** file on the Google colab (placed in afternoon_session folder)
- **Exercise:** Try to complete the script for the evaluation
  - You can get inspired by the Evaluator Class from the training point
  - Please note that class in another notebook used for evaluation on the validation set during training. Here you can figure out on how reusing it in order to evaluate a fine-tuned model.
  - Most of the Evaluator class code can be used.
- **If you had difficulties with the above approach,** try to evaluate your fine-tuned model inspired by the following implementation:
  - https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/ms_marco/eval_cross-encoder-trec-dl.py
- **Metrics to be report:** Map, P@1,5,10, Recall@10,100, and NDCG@10, 100
- Library for evaluation: pytrec_eval
- **Models that you evaluate:**
  - fine_tuned_model_path = 'your model'
  - fine_tuned_model_path = 'sentence-transformers/all-MiniLM-L12-v2'
  - fine_tuned_model_path = 'nlpaueb/legal-bert-base-uncased'
  - fine_tuned_model_path = './legal_bert_statuteretrieval_best_model'
- Finally, solution:

LLMs for statute reranking

- **First step:** Load **3_statute_reranking_with_llms_assignment.ipynb** file on the Google colab (placed in afternoon_session folder)
- The rest of information is available in notebook.
- You can continue this notebook later based on your own if you were interested.

Thanks