INFORMATION PROCESSING AND RETRIEVAL

INSTITUTO SUPERIOR TÉCNICO 2024/25

PROJECT v0.1

Acknowledgments: Wang, Lucy Lu, Kyle Lo, et al. 2020. "CORD-19: The COVID-19 Open Research Dataset." In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020. Online: Association for Computational Linguistics. `https://aclanthology.org/2020.nlpcovid19-acl.1/`.

1 Motivation

This project is motivated by the TREC-COVID Shared Task where users of the system are expected to be "Researchers, clinicians, and policy makers involved with the response to COVID-19 are constantly searching for reliable information on the virus and its impact." Consider the task of *ad-hoc retrieval*, given a *keyword query* providing a ranked list of possibly relevant documents from a given collection.

TREC-COVID Complete: Dataset used for TREC-COVID Shared Task

"TREC-COVID followed the TREC model for building IR test collections through community evaluations of search systems. The document set used in the challenge is the COVID-19 Open Research Dataset (CORD-19). This is a collection of biomedical literature articles that is updated regularly. Accordingly, TREC-COVID consisted of a series of rounds, with each round using a later version of the document set and a larger set of COVID-related topics. Participants in a round created ranked lists of documents for each topic ("runs") and submitted their runs to NIST. Based on the collective set of participants' runs, NIST created sets of documents to be assessed for relevance by human annotators with biomedical expertise. The results of the human annotation, known as relevance judgments, were then used to score the submitted runs. **The final document and topic sets together with the cumulative relevance judgments comprise a COVID test collection called TREC-COVID Complete.**"

This dataset is available via `ir_datasets` with the name `cord19/trec-covid`[1] (only includes article meta-data) and `cord19/fulltext/trec-covid`[2] (optional, includes article full texts).

- D: the *TREC-COVID Complete* (`cord19/trec-covid`), is a collection of 192,509 scientific articles related to COVID-19. It uses the `2020-07-16` version of the dataset, corresponding to the "complete" collection used for TREC COVID.
- Q: a set of 50 keyword queries, referred as *topics*, to be used to query the given collection. In addition to the actual query string "`title`", each topic also has `description` and `narrative`.
- R: relevance judgments (qrels), where each line in this dataset is a triplet (topic identifier $q \in Q$, document identifier $d \in D$, and multi-level relevance)

[1] `https://ir-datasets.com/cord19.html#cord19/trec-covid`
[2] `https://ir-datasets.com/cord19.html#cord19/fulltext/trec-covid`

Relevance level can be one of:

2 Relevant: the article is fully responsive to the information need as expressed by the topic, i.e. answers the Question in the topic. The article need not contain all information on the topic, but must, on its own, provide an answer to the question.

1 Partially Relevant: the article answers part of the question but would need to be combined with other information to get a complete answer.

0 Not Relevant: everything else.

For binary evaluation metrics consider all non-zero relevance levels as Relevant.

Primary goals

The target IR system should be developed in two steps:

1. in the first delivery, the IR system will accept topics as queries for ranking documents according to their metadata (e.g., title + abstract from the smaller metadata dataset `cord19/trec-covid`)
2. in the second delivery, students are allowed to use the fulltext to improve the ranking (i.e., body, from the larger `cord19/fulltext/trec-covid`).

Complementary notes

- the target IR system should be developed in Python. Students are free to use any available Python libraries to program the IR system, including ir_datasets, scikit-learn, spacy or nltk libraries;
- students can use external resources (including dictionaries, thesauri, ontologies, etc...) to guide text processing and scoring as long as their use is clearly identified;
- if needed, for the larger fulltext index you can opt to use whoosh, PyTerrier, etc;
- the training of the IR system on alternative text collections is considered to be out of the scope.

2 Project I (*first delivery*)

Topics: IR evaluation, indexing, Boolean and vector space models, ranking, and text processing.

Functionality

Program the following functions:

1. indexing(D,args)

@input	D and optional set of arguments on text preprocessing
@behavior	preprocesses each document in D and builds an efficient inverted index (with the necessary statistics for the subsequent functions)
@output	tuple with the inverted index I, indexing time and space required

2. boolean_query(q,k,I,args)

@input	topic q (identifier), number of top terms k, and index I
@behavior	uses the topic text to rank documents in the indexed collection using the Boolean IR model
@output	the filtered collection, specifically an ordered list of document identifiers

3. ranking(q,p,I,args)

@input	topic $q \in Q$ (identifier), number of top documents to return (p), index I, optional arguments on IR models
@behavior	uses the topic text to rank documents in the indexed collection using the tf.idf or BM25 retrieval model
@output	ordered set of top-p documents, specifically a list of pairs – (document identifier, scoring) – ordered in descending order of score

4. evaluation(Q,R,D,args)

@input	set of topics Q, document collection D, relevance judgments R
@behavior	evaluates the IR system ranking against the available relevance judgments
@output	evaluation statistics for the input ranking, including MAP, nDCG, P@10, and nDCG@10 (use standard tools from scikit-learn or even trec_eval)

In addition to the listed functions, students can further create other modular functionalities for an usable parameterization and execution of:

- *text processing options* (e.g. absence versus presence of phrase extraction);
- *IR models* (including the Boolean, TF-IDF, BM25, and LMs for IR).

IR system evaluation

Improvements on the IR system should be primarily guided by the metrics nDCG@10 and MAP.

The retrieved results under a Boolean model are assumed to be unordered and can be of arbitrary size. The retrieved results under weighted IR models are assumed to be ranked and, as reference, a fixed number of p=1000 documents should be considered for evaluation purposes.

A list of measures and/or performance curves should be provided for both the Boolean IR (simple retrieval) and weighted IR (ranking) settings, together with a brief rationale for their selection in the report.

Handling ranking differences

The multiplicity of options associated with the design of the target IR system (e.g. text processing and IR models) can lead to outputs with arbitrarily-high differences. **In the presence of multiple ranks**, CombSUM, CombMNZ or Reciprocal Rank Fusion (RRF) can be used for placing consensus. In this context, the ranking outputs produced under different parameters can be combined in accordance with the following equation:

$$\text{RRF\_score}(d \in D) = \sum_{f \in D} \frac{1}{50 + \text{rank}(f_d)}$$

where f is the set of top-ranked documents produced from a given IR system.

Under this score, you can assess whether consensus improves retrieval or, in alternative, there are specific processing options and retrieval models that yield best performance.

Questions to explore

Guidance on angles to conduct your analyzes and write your report:

1. Characterize the document collection D and topic collection Q. What is the distribution of informative terms in D before and after text processing? Are there overlapping top terms among Q topics?

2. Characterize the performance of the IR system. How much time and space are necessary for each stage – processing, indexing and retrieval – of the IR process?

3. Given a specific p, is the implemented IR system better at providing recall or precision guarantees?

4. Which p should be used by the IR system if the user has a preference towards: minimizing false positives, minimizing false negatives, or maximizing true positives?

5. Does performance strongly vary across topics in Q? Given the available relevance judgments, can we place a general consideration on the adequacy of the ranking model / retrieval function?

6. How different text processing and scoring options affect retrieval? Is reciprocal rank fusion useful to place ensemble decisions?

3 Project II (*second delivery*)

Topics: clustering, pagerank, IR evaluation

In this second stage of the IR system development, we will consider two major strategies in an attempt to improve its behavior. *First*, we will see whether the unsupervised organization of the corpus can offer guidance. *Second*, we will check whether modeling document interdependencies can yield improvements.

Grading: 25% clustering approach + 25% graph approach

3.1 Clustering approach: organizing collections

A discussion on how to use the knowledge produced by this clustering system to guide information retrieval is out the scope of this second delivery. As such, the focus should be primarily placed on finding a good and interpretable clustering solution.

Note: due to compute constraints, you can undertake this analysis in (`cord19/trec-covid/round1`) a subset of documents within the D collection. This uses the smaller "2020-04-10" collection version.

Functionalities to implement

1. clustering(D,args)

@input	document collection D (or topic collection Q), optional arguments on clustering analysis
@behavior	selects an adequate clustering algorithm and distance criteria to identify the best *number of clusters* for grouping the D (or Q) documents
@output	clustering solution given by a list of clusters, each entry is a cluster characterized by the pair (centroid, set of document/topic identifiers)

como escolher approach e distance:
- hierarchical, density, etc...

2. interpret(cluster,D,args)

@input	cluster and document collection D (or topic collection Q)
@behavior	describes the documents in the cluster considering both *median* and *medoid* criteria
@output	cluster description

3. evaluate(D,args)

@input	document collection D (or topic collection Q), optional arguments on clustering analysis
@behavior	evaluates a solution produced by the introduced clustering function
@output	clustering internal (and optionally external) criteria

Performance evaluation

como?

To evaluate the ability of organizing the corpus with clustering, appropriate internal measures should be considered to understand whether the documents (or topics) can be adequately separated.

Questions to explore

1. What is the (hypothesized) number of topic clusters? And document clusters in the D_{train} collection?
2. Are the clusters from previous? previous solutions cohesive? And well separated?
3. What the clustering of topic documents, Q, reveals regarding their conceptual organization and independence? Are there highly similar/overlapping topics?
4. Given a specific cluster of topics, check whether the medoid (a sort of prototype topic for the cluster) adequately represents the remaining topics in the given cluster.
5. How are the documents in the target collection organized? vamos decidir isto? Briefly discuss the importance of this information to understand the behavior of the target IR system.

3.2 Graph ranking approach: document centrality

Several previous studies have proposed to leverage graph centrality metrics in order to aid IR tasks. For instance, methods such as TextRank[3] bring principles from approaches similar to PageRank to static document collections.

Essentially, these methods are based on representing documents as a graph, where nodes correspond to documents, and where edges encode relationships between documents. For instance, an edge between two nodes can encode the similarity between the vector representations of two documents using the cosine measure.

In this context, students should develop a program that uses a PageRank-based approach for aiding the IR system.

Functionalities to implement

1. build_graph(D,sim,θ,args)

@input	document collection D, similarity criterion, and minimum similarity threshold θ
@behavior	computes pairwise similarities for the given document collection using the inputted similarity criterion; maps the pairwise relationships into a weighted undirected graph; and applies the θ threshold in order to remove edges with low similarity scores
@output	undirected graph capturing document relationships on the basis of their similarity

[3] http://web.eecs.umich.edu/ mihalcea/papers/mihalcea.emnlp04.pdf

2. undirected_page_rank(q,D,p,sim,θ,args)

@input	topic q, document collection D, number of top documents to return (p), and parameters associated with the graph construction and page rank algorithms
@behavior	given a topic query q, it applies a modified version of the page rank* prepared to traverse undirected graphs for document ranking
@output	ordered set of top-p documents – list of pairs (d, score) – in descending order of score

*candidate documents should be ranked according to a variation of the PageRank algorithm for undirected graphs, which computes a score for each candidate according to an iterative procedure based on the following equation:

$$\mathrm{PR}(d_i) = \frac{p}{N} + (1-p) \times \sum_{d_j \in \mathrm{Links}(d_i)} \frac{\mathrm{PR}(d_j)}{\mathrm{Links}(d_j)}$$

where $d_1, \ldots, d_N$ are the document candidates under consideration, $\mathrm{Links}(d_i)$ is the set of candidates that are similar to d_i (i.e., the set of nodes lined to d_i in the graph), and N is the total number of candidates. Notice that the PageRank definition considers a uniform residual probability for each node, usually set to $p = 0.15$.

The iterative procedure should be applied up to a maximum of 50 iterations.

You can use existing implementations for the PageRank algorithm (e.g. NetworkX package), but keep in mind that you should use a variant for unweighted graphs as shown in the previous equation.

The earlier introduced evaluation(Q,R,D,args) function in section 2, can be further extended to support the parameters of the target graph-based approach and return performance statistics.

Improving the graph-ranking method

The PageRank procedure from the previous exercise can be extended in order to consider a non-uniform prior probability for each candidate, and also in order to consider weighted edges in the graph, indicating the degree of association between the candidates. In this case, PageRank can be computed through the following iterative procedure:

$$\mathrm{PR}(d_i) = p \times \frac{\mathrm{Prior}(d_i)}{\sum_{d_j} \mathrm{Prior}(d_j)} + (1-p) \times \sum_{d_j \in \mathrm{Links}(d_i)} \frac{\mathrm{PR}(d_j) \times \mathrm{Weight}(d_j, d_i)}{\sum_{d_k \in \mathrm{Links}(d_j)} \mathrm{Weight}(d_j, d_k)}$$

In the equation, $\mathrm{Prior}(d_i)$ corresponds to a prior probability for node d_i, and $\mathrm{Weight}(d_i, d_j)$ corresponds to the weight of the edge between nodes d_i and d_j.

Student groups can be creative on how to place priors for the PageRank approach. One possibility is to consider document scores obtained using the original IR system developed in the first delivery. Nevertheless, students can explore alternative priors for documents.

Performance evaluation

Principles for evaluating the performance of the graph-enriched IR system are similar to the ones suggested for the original IR system in section 2.

In addition, comparisons can be established to assess the impact of the graph-based approach and how its performance varies in accordance with θ, the modeled document relationships, and the placed priors.

Questions to explore

- Does the graph ranking method based on document similarity aid IR?
  Quantify the differences in performance against the baseline IR system, and establish hypotheses on why is that so.
- How does ranking performance vary with θ?
- Upon the analysis of the graph, are there documents with higher graph centrality score? Which ones?
- Does the inclusion of non-uniform prior probabilities yield performance improvements? Hypothesize why is that so.

General guidelines

- **Deadlines** for project deliveries available at the course's webpage;
- **Grading** of project deliveries: 50% first delivery + 50% second delivery;
- Please always **consult the FAQ** on the course's webpage *before posting questions to your faculty hosts*. The subject of e-mails to your faculty should be preceded by [PRI project].
- **Delivery**: report (PDF), source code and Jupyter notebook demo highlighting major facilities;
- The **templates** for the reports and notebooks are listed in the course's webpage;
- We suggest **reports** to be organized in two parts: i) major decisions placed along the implementation of each task; and ii) point-by-point direct answering of the posed questions. Answers must be supported by quantitive empirical evidence;
- **Page limits** for each report are 8 main pages and 2 additional pages for supplementary results;
- Students should disclose the contribution of each member at the start of the project reports whenever efforts are unequal. *Marks are individual*;
- **Submissions**: Gxx.ZIP file via Fenix, where xx corresponds to your group number. The ZIP file should contain three files: Gxx_code.ZIP with your source code, Gxx_notebook.ipynb with your notebook demo, and Gxx_report.pdf with your report;
- Please note that it is possible to submit files several times on Fenix. Yet, only the last submission is stored in Fenix. You can submit versions ahead to prevent late-time problems.

Copy and plagiarism

- The project code and reports will be subjected to strict copy and plagiarism checkers;
- Checkers are run against other students (includes LLM-mediated content) and online content;
- If copy is detected after manual clearance for any of the above cases, the registration in PRI is nullified and IST guidelines will apply. These guidelines are also valid for students sharing the copied contents, irrespectively of the underlying intent.

Final remarks

- Please always **consult the FAQ** on the course's webpage **before posting questions to your faculty hosts**;
- The subject of e-mail contacts to your faculty hosts with project-related questions should be preceded by [PRI project] or [PRI essay].