Information Processing and Retrieval

Instituto Superior Técnico 2020

Project and Essay v0

Acknowledgments: National Institute of Standards and Technology (NIST), Reuters Ltd, Thomson Reuters, and TREC for providing the collection and associated relevance feedback to be used in this project.

1 Motivation

Consider the problem of providing a document of interest into a search engine to guide the retrieval of documents from a given collection.

This problem, formerly known as *filtering*, is a pervasive need nowadays, offering an effective means for *document navigation* and improving complex querying when topic descriptions are available.

Material

Consider the following material:

- D: the *Reuters Corpus Volume I* (RCV1) collection, a collection with over 800,000 manually categorized newswire stories made available by *Reuters Ltd.*. In this context, consider the two following sets:
  - D_{test}: the subset of documents within the D collection to be subjected to retrieval and evaluation purposes. The D_{test} collection, referred as testing collection, consists of all documents with dates posterior to September 30, 1996 (723,141 documents);
  - D_{train}: a subset of documents within the D collection to guide the behavior of the target Information Retrieval (IR) system in the presence of *relevance feedback*. This training collection consists of all documents with dates up to and including 30 September 1996 (83,650 documents);
- Q: a set of 100 documents, referred as *topics*, to be used as criteria to filter/query the given collection. Each document in the Q collection offers a topic description. 50 of the 100

topics were written by individuals, while the remaining 50 were artificially constructed based on the intersection of pairs of categories in RCV1;

- R: data containing relevance feedback (judgments), where each line in this dataset is a triplet (topic identifier $q \in Q$, document identifier $d \in D$, feedback) where feedback is either 1 when the document $d \in D$ is deemed as relevant for topic $q \in Q$ and 0 when non-relevant. Similarly to D, the relevance feedback is divided in two subsets:
  - R_{train}: containing the available relevance feedback for documents in D_{train};
  - R_{test}: containing the available relevance feedback for documents in D_{test}.

Primary goals

The target IR system will be developed in two steps:

(a) in the first delivery, the IR system will accept topics as queries for ranking documents in the D_{test} collection. Relevance feedback, R_{test}, is only allowed for evaluation purposes;

(b) in the second delivery, students are allowed to use the training collection D_{train}, together with the relevance feedback available for the documents in this collection, R_{train}, to improve the behavior of the original IR system.

Download

The aforementioned data is available at the course's webpage:

- rcv1.tar.xz corresponds to the D collection (access password sent via e-mail);
- topics.txt corresponds to the Q collection;
- qrels.train and qrels.test correspond to the R_{train} and R_{test} collections, respectively.

Complementary notes

- the target IR system should be developed in Python. Students are free to use available Python libraries to program the IR system, including scikit-learn, spacy or nltk libraries;
- the only fields (elements) of the document records in D that can be used are: newsitem, headline, text, dateline and byline. The category code fields must **not** be used;
- students can use external resources (including dictionaries, thesauri, ontologies, etc...) to guide text processing and scoring as long as their use is clearly identified;
- the training of the IR system on alternative text collections is considered to be out of the scope of the project.

2 Project I (*first delivery*)

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

Functionality

Program the following functions:

(a) 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

(b) extract_topic_query(q,I,k,args)

@input	topic $q \in Q$ (identifier), inverted index I, number of top terms for the topic (k), and optional arguments on scoring
@behavior	selects the top-k informative terms in q against I using parameterizable scoring
@output	list of k terms (a term can be either a word or phrase)

(c) boolean_query(q,k,I,args)

@input	topic q (identifier), number of top terms k, and index I
@behavior	maps the inputted topic into a simplified Boolean query using extract_topic_query and then search for matching* documents using the Boolean IR model
@output	the filtered collection, specifically an ordered list of document identifiers

*important: the Boolean querying should tolerate up to $round(0.2 \times k)$ term mismatches

(d) 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 directly the topic text (without extracting simpler queries) to rank documents in the indexed collection using the vector space model or probabilistic retrieval model
@output	ordered set of top-p documents, specifically a list of pairs – (document identifier, scoring) – ordered in descending order of score

(e) evaluation(Q_{test},R_{test},D_{test},args)

@input	set of topics $Q_{test} \subseteq Q$, document collection D_{test}, relevance feedback R_{test}, arguments on text processing and retrieval models
@behavior	uses the aforementioned functions of the target IR system to test simple retrieval (Boolean querying) tasks or ranking tasks for each topic $q \in Q_{test}$, and comprehensively evaluates the IR system against the available relevance feedback
@output	extensive evaluation statistics for the inputted queries, including recall-and-precision curves at different output sizes, MAP, BPREF analysis, cumulative gains and efficiency

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 and BM25 models).

IR system evaluation

The provided judgements, R, contain the documents that have been judged on a particular topic (indicating which are judged as relevant or not relevant). Any document not listed in R was not subjected to a judgment. As a result, students are free to ignore unjudged documents or treat unjudged documents as non-relevant. Still, the decision should be clearly stated upfront.

Performance improvements on the IR system should be primarily based on average uninterpolated precision. In addition, the F-β measure is also suggested. This measure, a function of recall and precision, uses the free β parameter to determine the relative weight of recall and precision. For this project, a value of β=0.5 has been chosen, corresponding to an emphasis on precision (β=1 is neutral).

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. Note, nevertheless, that precision and recall can be also assessed at different p thresholds.

In the context of ranking, the order of documents in ranked outputs must be carefully considered. Although we do not have a ground truth on the real ranks, some of IR measures – such as BPREF – consider whether relevant documents are ranked above irrelevant ones. BPREF is also robust to missing relevance judgments.

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:

(a) 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?

(b) 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?

(c) Considering the Boolean retrieval model, how k impacts the size of the retrieved document solutions? Should k be a fixed threshold or depend on the provided topic document (q-specific)?

(d) Given a specific p, is the implemented IR system better at providing recall or precision guarantees?

(e) 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?

(f) Does performance strongly vary across topics in Q? Given the available relevance feedback, can we place a general consideration on the adequacy of the ranking function?

(g) 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, classification, page ranking, IR evaluation

In this second stage of the IR system development, we will consider three 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 assess the aided value from relevance feedback. *Finally*, we will check whether modeling document interdependencies can yield improvements.

Grading: 25% clustering approach + 45% relevance feedback + 30% graph approach

3.1 Clustering approach: organizing collections

Forms of unsupervised corpus exploration, such as formal and coherent concept analysis, can be placed to guide IR tasks. In this project, we will consider an alternative approach – clustering – to this very end. The primary goal is to understand the organization of topics in the provided topic collection, as well as the organization of documents in the RCV1 collection.

In addition, the clustering system should provide a way of describing the documents from a given cluster using the cluster's centroid.

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. This is one of the challenges to be explored in the subsequent essay. As such, the focus should be primarily placed on finding a good and interpretable clustering solution.

Note: due to efficiency constraints, please undertake this analysis in D_{train} documents only, as opposed to clustering the full RCV1 collection, D.

Functionalities to implement

(a) 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)

(b) 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

(c) 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

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.

Students wanting to go an extra mile can further conduct an external evaluation (e.g. purity and revised rand index) on the document collection D using the document categories as a ground truth. To this end, please note that RCV1 documents have a category element from which we can extract this information.

Questions to explore

(a) What is the (hypothesized) number of topic clusters? And document clusters in the D_{train} collection?

(b) Are the clusters from previous solutions cohesive? And well separated?

(c) What the clustering of topic documents, Q, reveals regarding their conceptual organization and independence? Are there highly similar/overlapping topics?

(d) 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.

(e) How are the documents in the target collection organized? Briefly discuss the importance of this information to understand the behavior of the target IR system.

3.2 Supervised approach: incorporating relevance feedback

In the presence of relevance feedback, document filtering can be formulated as a supervised learning task. In this context, the IR system can be trained using documents in D_{train} collection and the available judgments for D_{train} (i.e. R_{train}).

Similarly to the first delivery, the testing sets, D_{test} and R_{test}, are provided to evaluate the behavior of the (trained) IR system.

Functionalities to implement

(a) training(q,D_{train},R_{train},args)

@input	topic document $q \in Q$, training collection D_{train}, judgments R_{train}, and optional arguments on the classification process
@behavior	learns a classification model to predict the relevance of documents on the topic q using D_{train} and R_{train}, where the training process is subjected to proper preprocessing, classifier's selection and hyperparameterization
@output	q-conditional classification model

(b) classify(d,q,M,args)

@input	document $d \in D_{test}$, topic $q \in Q$, and classification model M
@behavior	classifies the probability of document d to be relevant for topic q given M
@output	probabilistic classification output on the relevance of document d to the topic t

(c) evaluate(Q_{test},D_{test},R_{test},args)

@input	subset of topics $Q_{test} \subseteq Q$, testing document collection D_{test}, judgments R_{test}, and arguments for the classification and retrieval modules
@behavior	evaluates the behavior of the IR system in the presence and absence of relevance feedback. In the presence of relevance feedback, training and testing functions are called for each topic in Q_{test} for a more comprehensive assessment
@output	performance statistics regarding the underlying classification system and the behavior of the aided IR system

In addition to these core facilities, students should further implement the below described extensions of classifiers towards multiple IR models and ranking tasks.

Extension towards multiple IR models

The classification task is inherently prepared to handle data with arbitrary dimensionality. Some classifiers are internally able to handle high-dimensional data, while others commonly rely on dimensionality reduction procedures in order to select the most discriminative features.

In this context, the classification setting does not need to be restricted towards features gathered from a single IR model.

Extend the previous functionalities of the target classification system in order to allow the expansion of the feature space in the presence of multiple IR models. In this context, the individual term weights produced using different IR models can be integrated within a single feature space, including: i) term frequency, ii) inverse document frequency, iii) TF-IDF, and iv) BM25 features.

Students that want to go an extra mile can propose additional features of potential interest.

For a given topic $q \in Q$, you can now compare the performance of your classification system considering a feature space derived from a single retrieval model (e.g. term frequency) against the expanded feature space derived from multiple retrieval models.

Extension towards ranking

The previous classifiers are only able to determine whether a given document d is relevant or not for a given topic q. As such, they are as-is unable to rank documents.

Consider the two following strategies to extend the IR system to perform ranking:

- using the introduce classification system for removing non-relevant documents, followed by the application of traditional scoring criteria as given for instance by the application of cosine operator over the relevant documents;
- using the probabilistic outputs of the classification system, P(relevant—d,q), as a rough scoring criterion for ranking documents.

Select one of the introduced strategies for ranking purposes and briefly discuss its inherent merits and limitations.

Classification models

On the classifier selection: select and compare two classifiers. Some possibilities: i) Bayesian classifiers, such as naïve Bayes; ii) analogizers, such as kNN (k-Nearest Neighbor); iii) associative classifiers, such as random forests; and iv) neural networks, such as the multi-layer perceptron.

On the classifier hyperparameterization: we suggest that for, at least, one of the selected classifiers, you should attempt to optimize its parameters. Examples: number of neighbors and distance measure for kNN; number of trees and maximum depth for random forests; number of hidden layers and perceptrons per hidden layer for multi-layer perceptrons.

Your implementation can fully rely on packages with classification and hyperparameterization facilities, such as scikit-learn.

Performance evaluation

To assess the impact of relevance feedback on the target IR system, consider:

- comparing the performance of the IR system behavior for simple retrieval in the absence and presence of supervised learning from available judgments;
- evaluating the performance of the learned classifiers on the training and testing collections, D_{train} and D_{test}, in order to evaluate their overfitting propensity;
- comparing the ability of the IR system to rank documents in the absence and presence of relevance feedback (using one of the two strategies suggested to this end).

Questions to explore

(a) Does the incorporation of relevance feedback significantly impact the performance of the IR system? Hypothesize why is that so.

(b) Are performance improvements approximately uniform across topics? Are there topics substantially harder to classify? Which ones?

(c) From the tested classification variants (algorithms and parameterizations), which one yields better performance for simple retrieval? Hypothesize why is that so.

(d) Does the extension of the classification setting towards ranking yield significant performance improvements? Hypothesize why is that so considering results from rank-aware measures.

(e) Does the incorporation of additional features aid the behavior of the target IR system? Hypothesize why is that so.

3.3 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[1] 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

(a) 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

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

(b) 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 um 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_{test},R_{test},D_{test},args) function in section 2, can be further extended to support the parameters of the target graph-based approach and return performance statistics for a selected set of topics, Q_{test}.

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. In addition, different similarity functions to produce the edge weights can tested.

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 (without relevance feedback), 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.

4 Essay

In the second delivery, we explored three different ways on how to improve the behavior of the target IR system. The quest for the essay is to select **two** of the three assessed approaches and go deeper on the understanding of their relevance to IR by identifying practical principles to yield further performance improvements.

These principles, together with the intuition behind their importance for our IR system, should be enumerated in a direct, clear and practical way. The essay template is a *single* double-column page, i.e. one column per challenge.

Challenges

(a) **clustering system**:

- How can the (clustering-based) organization of the RCV1 collection guide simple retrieval? And ranking?
- How the clustering of topics can be used to guide the targeted filtering tasks?
- Can the application of clustering on the vocabulary (instead on the set of documents) be useful for IR? How?
- Do the aforementioned principles only impact IR efficacy? Can they be used to boost IR efficiency?

(b) **relevance feedback system**

- Half of the provided collection of topics were produced from the intersection of document categories. How can classifiers leverage on this knowledge to aid IR?
- How can classifiers be extended to distinguish text from different document elements (e.g. title and body fields)? Identify general principles on how classifiers may be able to underline terms from a given element (e.g. summary field).
- Lengthier documents are more verbose and therefore have higher chance of being judged as relevant for a randomly selected topic. If this is an undesirable bias to your IR system, how can the classification subsystem robustly handle this bias?

(c) **graph-based ranking system**

- If your original IR system is achieving good recall at the cost of a bad precision, which principles can be placed to revert this behavior?
- Are there alternatives to the introduced document graph that can yield improvements? Alternative centrality scores? Alternative priors?
- The built graphs are independent of the queried topics. Given a specific topic, how can this topic be used to infer a document graph more suitable to understand topic-conditional document similarities? How can such graph be used to guide IR tasks?

Essay evaluation criteria

- **Important remark**: It is neither necessary nor suggested the search for state-of-the-art literature to answer the selected subjects.

  The aim of this essay is to create a more *self-reflective and creative thinking* for the identification of principles to answer the selected challenges. The challenges for the essay are intentionally open, there is no single formula to tackle them.

- **Grading**: 50% challenge 1 + 50% challenge 2

- **Evaluation criteria**: clarity, adequacy, criativity, soundness and coverage of the answers to the selected challenges will be the major evaluation criteria of the essay. We are aware of the difficulty of conveying all your knowledge on this subject within a single page. Succinctness is part of the goal.

Essay guidelines

- **Individual submission**: In contrast with project deliveries, essay should be conducted and submitted *individually*.

- **Page limit**: the essay should have a maximum of 1-page text content. Text content beyond the first page will not be assessed. A second page with images and/or references is allowed.

- **Identical answers**: In contrast with project deliveries, we do not encourage extensive peer discussions. In addition to text plagiarism detectors, essays with identical answers to the same challenge (even for non-plagiarized text) can be discounted or nullified. Please check the evaluation section.

5 General guidelines

Submission

- **Deadlines** for project and essay deliveries available at the course's webpage;
- The **templates** for the project reports and essays are listed in the course's webpage;
- The **page limits** for each project delivery report are 4 pages and a single page for the essay;
- Grading of project deliveries: 15% first delivery + 20% second delivery (considering the 35% weight on the course's grade). Grading will be primarily based on the report quality, please do not neglect the amount of time required to produce a good report;
- Students should disclose the contribution of each member at the start of the project reports whenever efforts are unequal. *Marks are individual*;
- Project deliveries in *group*. Essay deliveries are *individual*;
- First and second **project submissions**: Gxx.ZIP file via Fenix, where xx corresponds to your group number. The ZIP file should contain two further files: Gxx_code.ZIP with your source code, and Gxx_report.pdf with your two-page report;
- **Essay submissions**: istxxxxxx.PDF file via Fenix, where xxxxxx corresponds to your ist number;
- Please note that it is possible to submit files several times on Fenix. As such, you can submit preliminary versions in advance to prevent late-time problems. Yet, note that only the last submission will be considered for evaluation.

Copy and plagiarism

- The project code and project reports will be subjected to strict copy checkers;
- The essay submissions will be subjected to strict plagiarism checkers among deliveries and against online and published content. Please also note that essays should be conducted individually (check remarks on essays with identical answers);
- 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 code, independently 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].