INFORMATION PROCESSING AND RETRIEVAL

INSTITUTO SUPERIOR TÉCNICO 2023/24

PROJECT v0

Consider the IR tasks of **keyword extraction** and **text summarization**. For extracting keywords from a document, some IR systems rank the relevance of normalized noun phrases in a document. For summarizing text, IR systems commonly follow *abstractive* or *extractive* approaches. Extractive approaches select the most informative sentences from the original document, and produce a summary where these sentences are ordered by relevance or location in the original document. The fundamental difficulty of these tasks lie in identifying and ranking relevant text that capture the main ideas behind a document, while ensuring minimum redundancy.

Material

BBC News Summary repository contains the document collection and corresponding summaries.

Document collection. The corpus is composed of 2225 journalistic texts collected from the online newspaper BBC News, $D = \{d_1, .., d_{2225}\}$, along five categories: business, entertainment, politics, sports and tech news, $D = D_{business} \cup D_{entertainment} \cup D_{politics} \cup D_{sports} \cup D_{tech}$.

category	**size**	**document** (avg #tokens/text)	**summary** (avg #tokens/text)
business	510	328.88 ± 135.8	139.93 ± 59.4
entertainment	386	330.62 ± 261.5	144.05 ± 124.0
politics	417	453.97 ± 299.8	195.71 ± 139.6
sports	511	329.26 ± 187.8	143.19 ± 80.8
tech	401	502.70 ± 239.6	213.84 ± 111.4

Table 1: BBC News Summary: essential statistics

Reference summaries. The corpus contains the corresponding summaries, $R = \{r_1, .., r_{2225}\}$, also termed extracts, obtained using a state-of-the-art extractive method[1]. There is no need to be familiar with this method for the current project. Note that, although reference extracts may not correspond to the optimal summaries, they offer a stable reference ground for summarization.

Download. *BBC News* repository is temporarily made available at: `https://fenix.tecnico.ulisboa.pt/downloadFile/3096843219121707/BBC%20News%20Summary.zip` (4.3Mb)

Acknowledgments. BBC (`http://mlg.ucd.ie/datasets/bbc.html`), Kaggle (`https://www.kaggle.com/datasets/pariza/bbc-news-summary`), D. Greene, P. Cunningham, Pariza Sharif

[1] Kernel text subspace clustering method that ensures both centrality and non-redundancy: *Greene and Cunningham. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering, ICML 2006.*

Project goals

The target IR system will be developed in two steps:

1. in the first delivery, keyword extraction and summarization will be done unsupervisedly over the D collection. Reference extracts are uniquely used to guide evaluation;
2. in the second delivery, we will explore the structure of the corpus and use the reference extracts as feedback to guide the summarization process.

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 the summarization tasks; 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.

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;
- students can use external resources (including dictionaries, thesauri, ontologies, etc.) to guide text processing and scoring as long as their use is clearly identified;
- students can implement advanced summarization solutions provided in Kaggle or described in scientific literature. Yet, as the project will be solely evaluated on the ability to answer the requirements and questions in this statement, such direction is discouraged.

Project I (*first delivery*)

Topics: IR evaluation, IR models, indexing, ranking, text processing

Task: On the first part of the project, students should be able to establish a sound solution for the keyword extraction and summarization tasks using *classic IR querying* principles.

Grading: **50%** sound-functional IR system (5% indexing + 5% keyword extraction + 20% summarization basis + 10% evaluation + 5% consensus + 5% redundancy) + **50%** question answering

Functionality

Program the following functions:

1. indexing(D,args)

@input	document collection D and optional arguments on text preprocessing
@behavior	preprocesses the collection and, using existing libraries, builds an inverted index with the relevant statistics for the subsequent summarization functions
@output	pair with the inverted index I and indexing time

2. summarization(d,p,l,o,I,args)

@input	document d, maximum number of sentences (p) and/or characters (l), order of presentation o (appearance in text *vs* relevance), inverted index I or the collection D, and optional arguments on IR models
@behavior	preprocesses d, assesses the relevance of each sentence in d against I according to args, and presents them in accordance with p, l and o
@output	summary s of document d, i.e. ordered pairs $(\text{sentence position in } d, \text{score})$

3. keyword_extraction(d,p,I,args)

@input	document d, maximum number of keywords p, inverted index I, and optional arguments on IR model choices
@behavior	extracts the top informative p keywords in d against I according to args
@output	ordered set of p keywords

4. evaluation(S_{set},R_{set},args)

@input	the set of summaries S_{set} produced from selected documents $D_{set} \subseteq D$ (e.g. a single document, a category of documents, the whole collection), the corresponding reference extracts R_{set}, and optional arguments (evaluation, preprocessing, model options)
@behavior	assesses the produced summaries against the reference ones using the target evaluation criteria
@output	evaluation statistics, including F-measuring at predefined p-or-l summary limits, recall-and-precision curves, MAP, and efficiency

Design options

Multiple solutions can be envisioned to answer the summarization and extraction tasks. Term-and-document frequencies can be assessed at **sentence**, **document**, or **collection** levels. Your choices will affect the behavior of (1) function and performance of (2) and (3) tasks. As such choices are a central part of this delivery, your faculty hosts are unable to offer definite answers on the best solution. Get creative and, if you are pursuing more than one solution per task, compare the accuracy of the tested solutions. Otherwise, provide the rationale for your choice.

Text processing

Your vector spaces should not only include words but also noun phrases. Assessing the impact of alternative **text processing options**, such as stop word removal or lemmatization *vs* stemming choices, is considered to be out of the scope of the project. Place preprocessing decisions, motivate their choice, and fix them throughout the project.

IR models

Your *extraction* approach should be primarily grounded on **TF-IDF** criteria using the inverted index I. The *summarization* approach should further support **BM25** using the inverted index I and **BERT embeddings** using the collection D. BM25 should be considered with the default parameters, k_1=1.2 and b=0.75. Do not forget that the average length parameter in BM25 can correspond to sentence or document level stances depending on the envisioned strategy. No fine-tuning is required for the acquisition of BERT embeddings.

IR system evaluation

Keywords can be manually inspected for a few selected documents.

Sound performance evaluation is only necessary for the summarization task. The behavior of the summarization system should be primarily assessed based on average uninterpolated precision. F-β measurements are also suggested for a pre-fixed summary length. For this project, we suggest fixing a value β to offer an emphasis on precision over recall. As reference, length parameters can be fixed as p=8 and l=500. Note nevertheless that precision-recall curves should not depend on these reference bounds.

Evaluation should be conducted for a specific selected document, category, and the overall collection. For the later cases, multiple estimates are acquired, thus particular attention should be placed to the presentation of results acquired from document sets by selecting adequate statistics and chart visualizations.

Statistical comparisons should be undertaken to assess the performance of the different IR models, preprocessing options and, if available, summarization solutions. Finally, the ability to summarize documents can be assess for different categories.

Handling ranking differences (10% in total)

The diversity of options on the design of the target IR system can lead to arbitrarily-high differences on the produced summaries. In the presence of multiple keyword ranks or sentence ranks, the Reciprocal Rank Fusion (RRF) can be used to place consensus,

$$\mathrm{RRF}(s) = \sum_{f \in F} \frac{1}{\mu + \mathrm{rank}(f(s))},$$

where F is the set of IR options (e.g. summarization with BM25 and BERT), μ is a constant that can be fixed at μ=5, s is the ranked object (e.g. a sentence in the context of summarization), and $rank(f(s))$ is rank for s using the IR behavior f (e.g. rank of sentence s using BM25).

We ask you to assess the impact RRF-based consensus on the *summarization* task only. In particular, whether the *consensus* from BM25 and BERT sentence ranks improves performance or, otherwise, whether vector space models or language models alone yield better performance.

Handling redundancy (10% in total)

A notable problem of all previous solutions is the potential high redundancy among the selected keywords or sentences. In the context of summarization, the paper entitled "*The use of MMR: diversity-based reranking for reordering documents and producing summaries*", Carbonell and Goldstein propose Maximal Marginal Relevance (MRR) to increase sentence diversity. The proposed method iteratively selects the most informative sentence, adding it to the set of sentences in the summary and removing it from the document. Next sentence selection is based on the MMR score that simultaneously attempts to select sentences that are relevant and dissimilar to the sentences selected so far (non-redundancy),

$$\mathrm{MMR}(s) = (1 - \lambda) \times sim(s, d) - \lambda \sum_{v \in S} sim(s, v). \tag{1}$$

where S is the sentences composing the summary at a given time and sim is the cosine similarity.

We ask you to augment your summarization system with MMR. Selecting one IR model and λ=0.5, assess the impact of MMR against the previous redundancy-unaware summarization system on the overall collection D. Further pick an arbitrary document, and further assess the differences produced by varying λ.

Questions to explore in the report

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

1. Describe the corpus D and summaries S. Are terms uniformly distributed regarding TF-IDF?
2. How does the summarization system perform for the full collection?
   And within each category? Any intuition for the observed differences?
3. How IR models affect summaries? How vector space models compare with language models?
4. Is Reciprocal Rank Fusion (RRF) useful to aid decisions?
5. Considering MMR, how λ impacts the accuracy (against ideal extracts) of summaries?
   Should λ be a fixed threshold or depend on the provided topic document (d-specific)?
6. At the suggested p length threshold, is the system better at promoting recall or precision?

Project II (*second delivery*)

Topics: clustering and classification

Task: In this second stage of the IR system development, we will consider two major strategies:

- assess whether clustering can aid the feature extraction and summarization tasks
- assess the added value of the relevance feedback acquired from the reference extracts

Grading: 50% clustering approach + 50% relevance feedback

A) Unsupervised IR

Sentence clustering can be applied to identify dissimilar topics within a document, thus ensuring the extracted keywords and summaries from a document cover all relevant topics with minimal redundancy. To assess the role of clustering for these tasks, five major steps are suggested:

- learn adequate feature spaces or BERT embeddings for the subsequent clustering task;
- placing adequate clustering choices for the target tasks, identify an adequate number of topics, paying to attention to the fact that not all clusters may be relevant;
- design a strategy to select prototype sentences or keywords from candidate clusters; and
- answer the target tasks, assessing the differences in summarization efficacy against the baseline IR system developed in the first delivery.

Functionality

1. sentence_clustering(d,I,args)

@input	document d, optional inverted index I, optional clustering args
@behavior	identifies the best number of sentence clusters for the target tasks according to proper internal indices, returning the corresponding clustering solution
@output	clustering solution C

2. summarization(d,C,I,args)

@input	document d, sentence clusters C, optional inverted index I and guiding args
@behavior	ranks non-redundant sentences from the clustering solution, taking into attention that ranking criteria can be derived from the clusters' properties
@output	summary (without pre-fixed size limits)

3. keyword_extraction(d,C,I,args)

@input	document d, sentence clusters C, optional inverted index I and guiding args
@behavior	extracts non-redundant keywords from the clustering solution, taking into consideration that multiple clusters may share high-relevant terms
@output	set of keywords (without pre-fixed cardinality)

4. evaluation(S_{set},R_{set},args) – from Project I

Questions to explore

1. Do clustering-guided summarization significantly alters the behavior and efficacy of the IR system? Hypothesize why is this so.
2. How sentence representations, clustering choices, and rank criteria impact summarization?
3. Are anchor sentences (capturing multiple topics) included? And less relevant outlier sentences excluded? Justify.
4. Given a set of documents, plot the distribution of the number of keywords per document. Are keywords generally dissimilar? If not, how would you tackle this challenge?

B) Supervised IR using reference summaries

In the presence of relevance feedback from reference extracts, summarization can be formulated as a supervised learning-to-rank task. In this case, the sentences from a given document can be represented through a set of descriptive features (e.g., position in the paragraph, ranking scores), and a learned mapping between the features of a sentence and its presence or absence in the reference summary. To this end, the summarization system can be trained using sentences from documents in a dedicated D_{train} collection and the corresponding reference extracts, R_{train}.

Functionalities to implement

1. feature_extraction(s,d,args)

@input	sentence s and the enclosed document d
@behavior	extracts features of potential interest considering the characteristics of the sentence and of the wrapping document
@output	sentence-specific feature vector

2. training(D_{train},R_{train},args)

@input	training document collection D_{train}, reference extracts R_{train}, and optional arguments on the classification process
@behavior	learns a classifier to predict the presence of a sentence in the summary
@output	classification model

3. summarization(d,M,p,l,args)

@input	document $d \in D_{test}$, classification model M, maximum number of sentences (p) and/or characters (l), and guiding args
@behavior	using M, summarizes d without prefixed size limits by identifying relevant sentence candidates for the summary or, in alternative, with prefixed size limits (p and/or l) by ranking the sentences s in d based on their likelihood to be selected as part of the document summary
@output	summary s of document d

4. supervised_evaluation(D_{test},R_{test},M,args)

@input	testing document collection D_{test}, corresponding reference extracts R_{test}, the learnt model M, and guiding evaluation args
@behavior	evaluates the behavior of the given classifier using (3)
@output	confusion-based scores (e.g. precision, recall, AUC) of the M classifier

5. evaluation($S_{set} \subseteq S_{test}$,$R_{set} \subseteq R_{test}$,args) – from Project I

Feature extraction

With the aim of describing sentences, the following features should be considered: i) position of the sentence in the enclosed paragraph and document; ii) cosine similarity of the sentence against document contents using TF-IDF, BM25 and BERT.
Students wanting to go an extra mile can explore and propose other features of potential interest.

Performance evaluation

To assess the impact of reference extracts on the target IR system, compare the performance of the summarization system behavior in the absence and presence of feedback from reference extracts recovering IR evaluation gathered in the first delivery.

To this end, ensure that the selected documents for this evaluation do not belong to the set of documents used to trained the summarization model.

Classification models

Select and compare the performance of two classifiers. Possibilities: i) Bayesian classifiers, such as naïve Bayes; ii) analogizers, such as kNN (k-Nearest Neighbor); iii) associative classifiers, such as random forests or XGBoost; or iv) neural networks, such as a multi-layer perceptrons. The hyperparameterization of the selected classifiers is optional. Your implementation can fully rely on packages with classification and hyperparameterization facilities, such as scikit-learn.

Questions to explore

1. Does the incorporation of relevance feedback from ideal extracts significantly impact the performance of the IR system? Hypothesize why is that so.
2. Are the learned models able to generalize from one category to another? Justify.
3. Which features appear to be more relevant to the target summarization task? Do sentence-location features aid summarization?
4. In alternative to the given reference extracts, consider the presence of manual abstractive summaries, can supervised IR be used to explore such feedback? Justify.

END