Università
della
Svizzera
italiana

**Facoltà
di
scienze
informatiche**

Introduction to PyTerrier

**By Federico Ravenda
and
Prof. Crestani & Landoni**

federico.ravenda@usi.ch

Information Retrieval
Fall Semester, 2023

Agenda

- ❑ What is PyTerrier?
- ❑ Advantages of PyTerrier
- ❑ Who uses PyTerrier?
- ❑ Process of Indexing
- ❑ Retrieving Documents

Links

PyTerrier repository

- https://github.com/terrier-org/pyterrier

PyTerrier documentation

- https://pyterrier.readthedocs.io

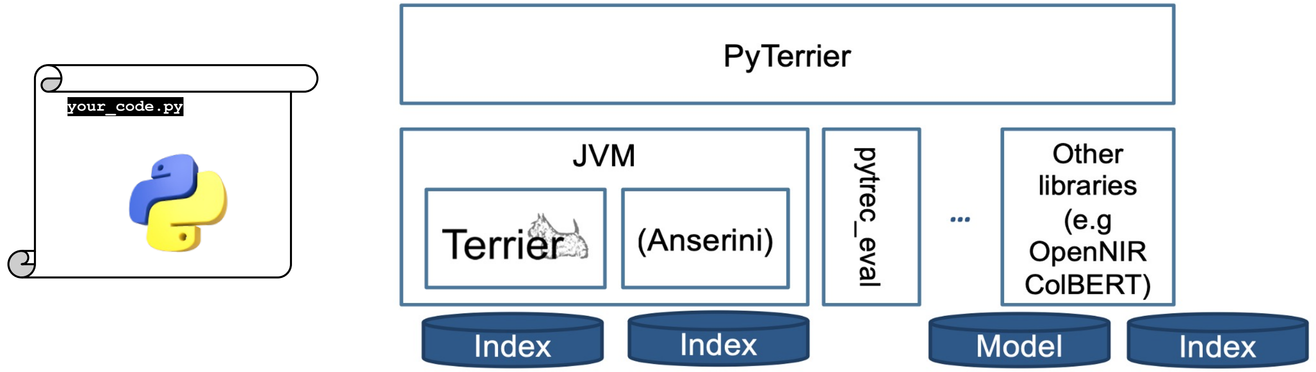
Tutorial repository

- colab

Reference: https://github.com/terrier-org/ecir2021tutorial

What is PyTerrier

- A Python layer above Terrier (and *other* IR platforms). Provides a high-level API for indexing and retrieval tasks. While PyTerrier was new in 2020, Terrier is written in Java and has a long history dating back to 2001.
- A Python framework for expressing and evaluating IR experiments.
- Bridges the gap between Python's ease-of-use and Terrier's powerful.



Why PyTerrier?

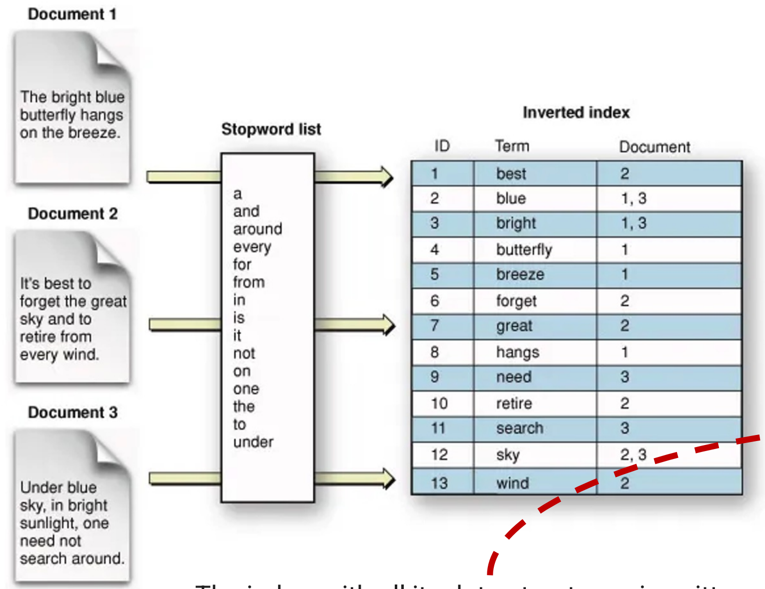
- ❑ **Python Integration**: PyTerrier is a Python-based library, making it a natural choice for those already familiar with Python.
- ❑ **Research & Experimentation**: Designed primarily for experimental information retrieval research. Ideal for testing new algorithms and approaches.
- ❑ **Ease of Use**: Given our familiarity with Python, PyTerrier offers a more intuitive and developer-friendly environment.
- ❑ **Customization**: Being Python-based, it's easier to customize and tweak according to specific research needs.
- ❑ **Learning Curve**: For a class already versed in Python, the learning curve for PyTerrier is significantly reduced compared to diving into a new ecosystem like Solr.
- ❑ **Community & Support**: Growing community of researchers and developers focused on Python-based information retrieval.

Advantages of PyTerrier

- Integration with Python Ecosystem:
  - Easily integrates with popular Python libraries.
- Efficient Indexing:
  - Provides fast and scalable indexing solutions.
- Flexible Retrieval Models:
  - Supports various state-of-the-art retrieval models.
- Rich Experimentation:
  - Facilitates systematic experimentation and evaluation.
- Community-driven:
  - Actively developed with an increasing number of contributors.

Indexing with PyTerrier

PyTerrier makes it easy to index standard Python data structures, particularly Pandas dataframes


Document 1
The bright blue butterfly hangs on the breeze.
Document 2
It's best to forget the great sky and to retire from every wind.
Document 3
Under blue sky, in bright sunlight, one need not search around.
Stopword list
a
and
around
every
for
from
in
is
it
not
on
one
the
to
under
Inverted index

ID	Term	Document
1	best	2
2	blue	1, 3
3	bright	1, 3
4	butterfly	1
5	breeze	1
6	forget	2
7	great	2
8	hangs	1
9	need	3
10	retire	2
11	search	3
12	sky	2, 3
13	wind	2

```
docs_df = pd.DataFrame([
        ["d1", "the bright blue butterfly hangs on the breeze"],
        ["d2", "it is best to forget the great sky and to retire from every wind"],
        ["d3", "under blue sky in bright sunlight one need not search around"]
    ], columns=["docno", "text"])

docs_df
```

docno	text
d1	the bright blue butterfly hangs on the breeze
d2	it is best to forget the great sky and to reti...
d3	under blue sky in bright sunlight one need not...

Data to index

```
indexer = pt.DFIndexer("./index_3docs", overwrite=True)
index_ref = indexer.index(docs_df["text"], docs_df["docno"])
```

The index, with all its data structures, is written into a directory called index_3docs.

Indexing with PyTerrier

You can access the file data.properties to explore some of the stats of your collection that was indexed

```
print(index.getCollectionStatistics().toString())
```

```
Number of documents: 3
Number of terms: 14
Number of postings: 17
Number of fields: 0
Number of tokens: 17
Field names: []
Positions:   false
```

```
 for kv in index.getLexicon():
  print("%s  -> %s " % (kv.getKey(), kv.getValue().toString()  ))
```

- **Nt** is the number of unique documents that each term occurs in – this is useful for calculating IDF
- **TF** is the total number of occurrences – some weighting models use this instead of Nt
- The numbers in the @{} are a pointer – they tell Terrier where the postings are for that term in the inverted index data structure.

```
best  -> term6 Nt=1 TF=1 maxTF=1 @{0 0 0}
blue  -> term0 Nt=2 TF=2 maxTF=1 @{0 0 4}
breez  -> term1 Nt=1 TF=1 maxTF=1 @{0 1 2}
bright  -> term3 Nt=2 TF=2 maxTF=1 @{0 1 4}
butterfli  -> term2 Nt=1 TF=1 maxTF=1 @{0 2 2}
forget  -> term5 Nt=1 TF=1 maxTF=1 @{0 2 4}
great  -> term8 Nt=1 TF=1 maxTF=1 @{0 3 0}
hang  -> term4 Nt=1 TF=1 maxTF=1 @{0 3 4}
need  -> term12 Nt=1 TF=1 maxTF=1 @{0 3 6}
retir  -> term9 Nt=1 TF=1 maxTF=1 @{0 4 2}
search  -> term13 Nt=1 TF=1 maxTF=1 @{0 4 6}
sky  -> term7 Nt=2 TF=2 maxTF=1 @{0 5 2}
sunlight  -> term11 Nt=1 TF=1 maxTF=1 @{0 6 0}
wind  -> term10 Nt=1 TF=1 maxTF=1 @{0 6 4}
```

Facoltà
di
scienze
informatiche

Retrieval

After getting the index, we can perform retrieval on it. We use the **BatchRetrieve** Class

```
br = pt.BatchRetrieve(index, wmodel="Tf") #Alternative Models: "TF_IDF", "BM25"
br.search("bright")
```

And **many** more…

	qid	docid	docno	rank	score	query
0	1	0	d1	0	1.0	bright
1	1	2	d3	1	1.0	bright

A very naive example!

`search()` method returns:

- `qid`: this is by default "1", since it's our first and only query
- `docid`: Terrier' internal integer for each document
- `docno`: the external (string) unique identifier for each document
- `score`: since we use the `Tf` weighting model, this score corresponds the total frequency of the query (terms) in each document
- `rank`: A handy attribute showing the descending order by score
- `query`: the input query

http://terrier.org/docs/current/javadoc/org/terrier/matching/models/package-summary.html

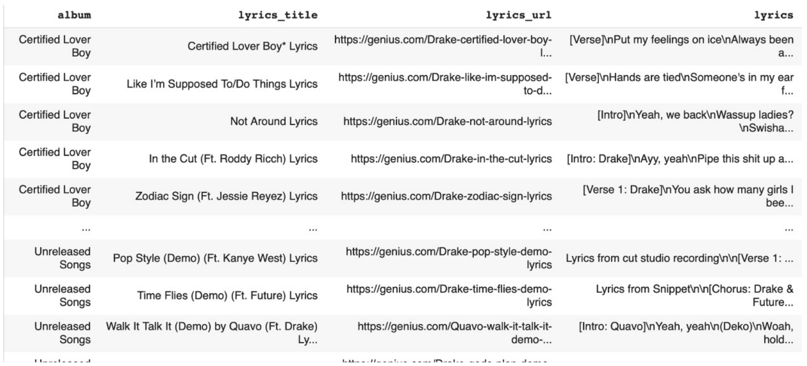
Facoltà
di
scienze
informatiche

Retrieval - A more complex Example

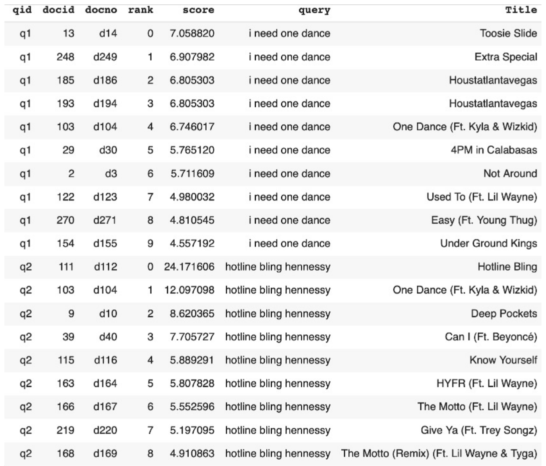
```
bm25 = pt.BatchRetrieve(index, wmodel="BM25")
queries = pd.DataFrame([["q1", "i need one dance"], ["q2", "hotline bling hennessy"]], columns=["qid",
"query"])
bm25.transform(queries)
```

More than a Query at a Time

```
pt.io.write_results(results, "res_bm25.txt", format='trec') # saving the results in trec_format
```



album	lyrics_title	lyrics_url	lyrics
Certified Lover Boy	Certified Lover Boy* Lyrics	https://genius.com/Drake-certified-lover-boy-l...	[Verse]\nPut my feelings on ice\nAlways been a...
Certified Lover Boy	Like I'm Supposed To/Do Things Lyrics	https://genius.com/Drake-like-im-supposed-to-d...	[Verse]\nHands are tied\nSomeone's in my ear f...
Certified Lover Boy	Not Around Lyrics	https://genius.com/Drake-not-around-lyrics	[Intro]\nYeah, we back\nWassup ladies? \nSwisha...
Certified Lover Boy	In the Cut (Ft. Roddy Ricch) Lyrics	https://genius.com/Drake-in-the-cut-lyrics	[Intro: Drake]\nAyy, yeah\nPipe this shit up a...
Certified Lover Boy	Zodiac Sign (Ft. Jessie Reyez) Lyrics	https://genius.com/Drake-zodiac-sign-lyrics	[Verse 1: Drake]\nYou ask how many girls I bee...
...	...	...	...
Unreleased Songs	Pop Style (Demo) (Ft. Kanye West) Lyrics	https://genius.com/Drake-pop-style-demo-lyrics	Lyrics from cut studio recording\n\n[Verse 1: ...
Unreleased Songs	Time Flies (Demo) (Ft. Future) Lyrics	https://genius.com/Drake-time-flies-demo-lyrics	Lyrics from Snippet\n\n[Chorus: Drake & Future...
Unreleased Songs	Walk It Talk It (Demo) by Quavo (Ft. Drake) Ly...	https://genius.com/Quavo-walk-it-talk-it-demo-...	[Intro: Quavo]\nYeah, yeah\n(Deko)\nWoah, hold...



qid	docid	docno	rank	score	query	Title
q1	13	d14	0	7.058820	i need one dance	Toosie Slide
q1	248	d249	1	6.907982	i need one dance	Extra Special
q1	185	d186	2	6.805303	i need one dance	Houstatlantavegas
q1	193	d194	3	6.805303	i need one dance	Houstatlantavegas
q1	103	d104	4	6.746017	i need one dance	One Dance (Ft. Kyla & Wizkid)
q1	29	d30	5	5.765120	i need one dance	4PM in Calabasas
q1	2	d3	6	5.711609	i need one dance	Not Around
q1	122	d123	7	4.980032	i need one dance	Used To (Ft. Lil Wayne)
q1	270	d271	8	4.810545	i need one dance	Easy (Ft. Young Thug)
q1	154	d155	9	4.557192	i need one dance	Under Ground Kings
q2	111	d112	0	24.171606	hotline bling hennessy	Hotline Bling
q2	103	d104	1	12.097098	hotline bling hennessy	One Dance (Ft. Kyla & Wizkid)
q2	9	d10	2	8.620365	hotline bling hennessy	Deep Pockets
q2	39	d40	3	7.705727	hotline bling hennessy	Can I (Ft. Beyoncé)
q2	115	d116	4	5.889291	hotline bling hennessy	Know Yourself
q2	163	d164	5	5.807828	hotline bling hennessy	HYFR (Ft. Lil Wayne)
q2	166	d167	6	5.552596	hotline bling hennessy	The Motto (Ft. Lil Wayne)
q2	219	d220	7	5.197095	hotline bling hennessy	Give Ya (Ft. Trey Songz)
q2	168	d169	8	4.910863	hotline bling hennessy	The Motto (Remix) (Ft. Lil Wayne & Tyga)

Università
della
Svizzera
italiana

**Facoltà
di
scienze
informatiche**

Thank You for the Attention & Good Luck for the Project!