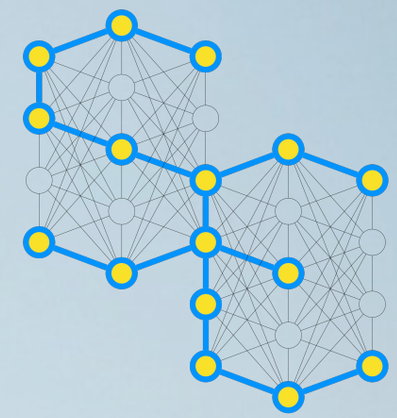


IR Basics Hands-on

Search Engines

Master Degree in Computer Engineering
Master Degree in Data Science
Academic Year 2022/2023

Nicola Ferro

Intelligent Interactive Information Access (IIIA) Hub
Department of Information Engineering
University of Padua

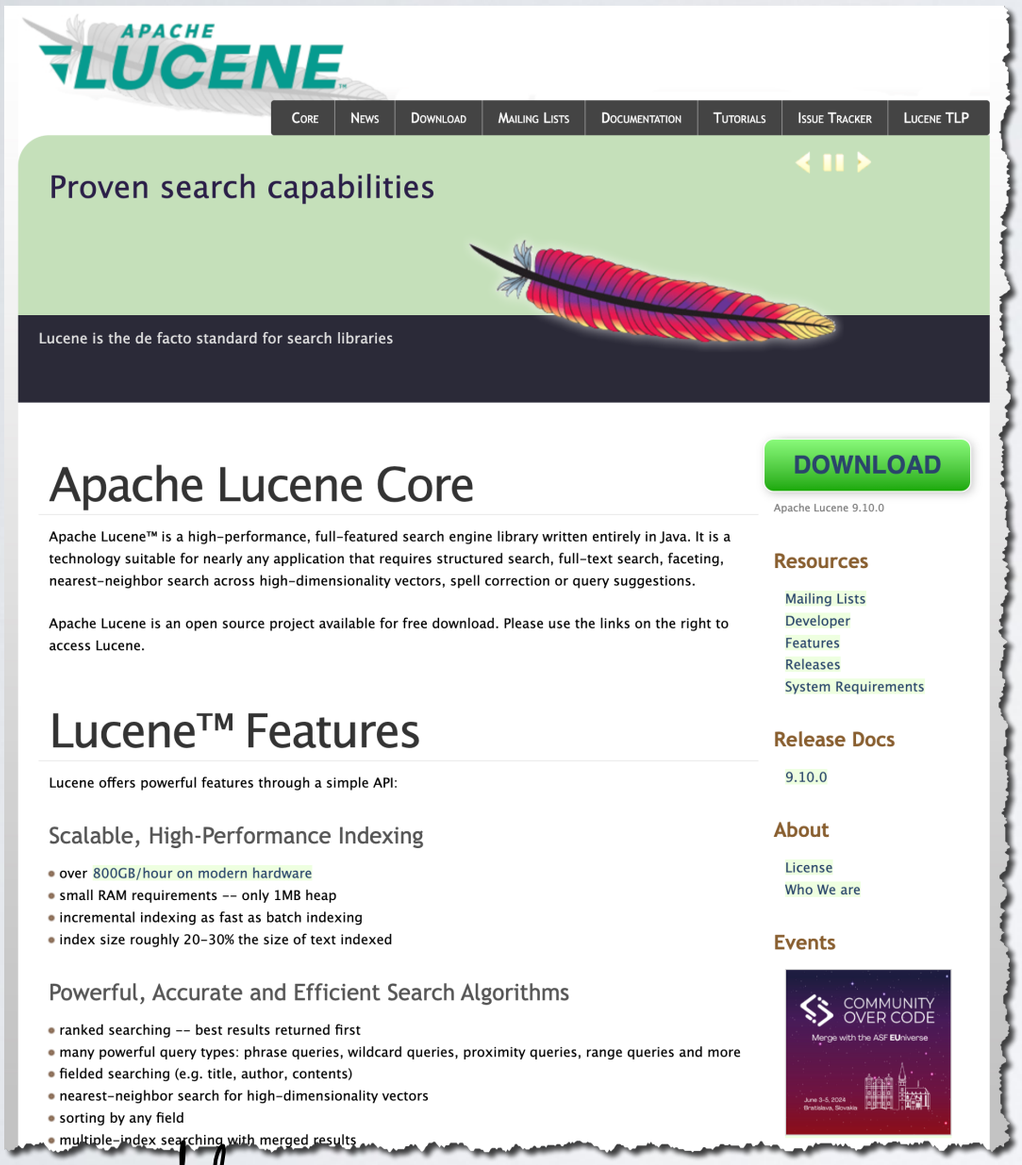
Outline

- Apache Lucene
- Hello, IR!
- Character Encoding
- Hello, TIPSTER!

Apache Lucene

Apache Lucene

https://lucene.apache.org/core/

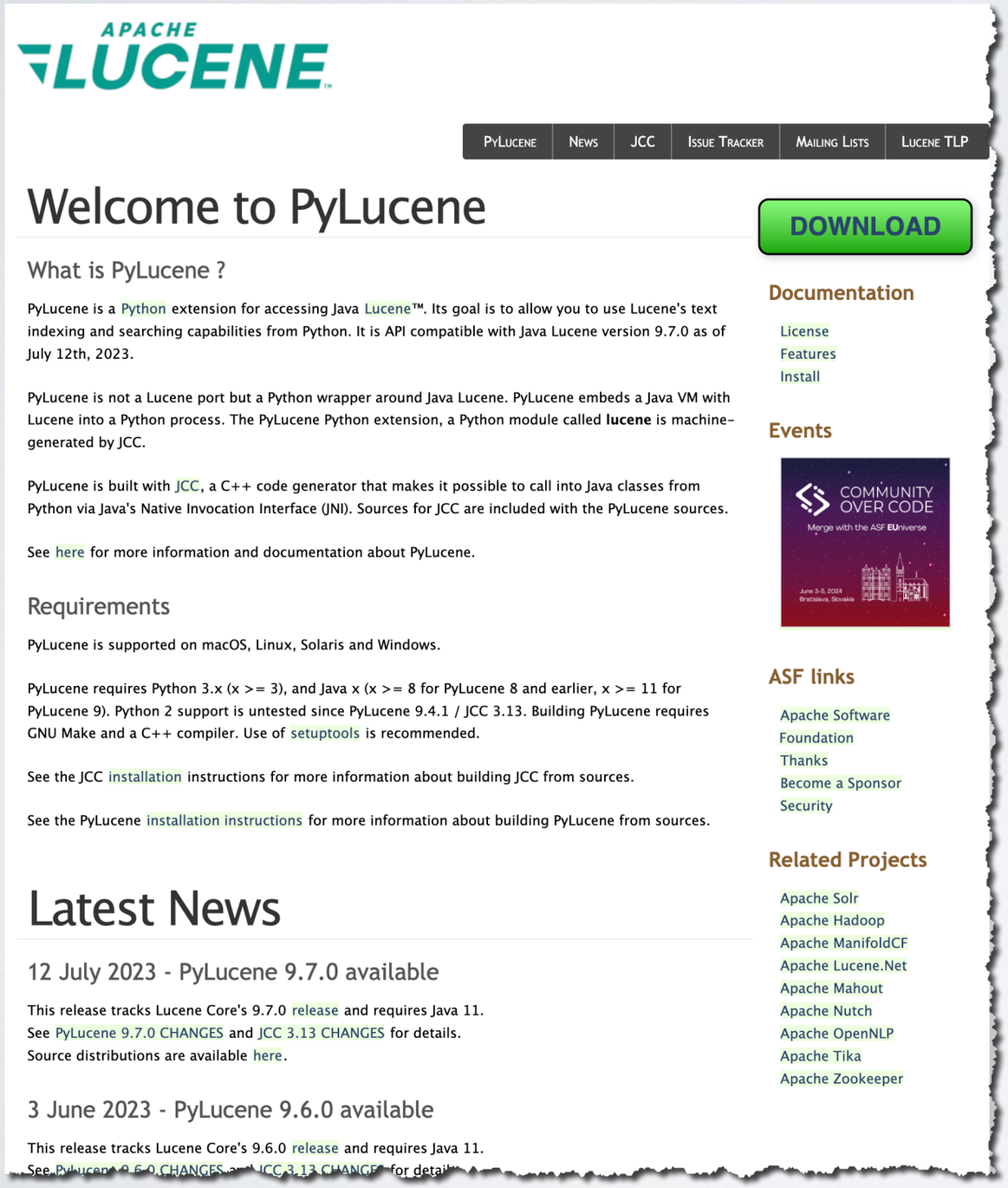


progetto in Java

- Lucene is a powerful open-source Java library for indexing and searching
- You can download both binaries and **source code**
  - Always keep the source code at hand and look at it
- It has an extensive **documentation**
  - Always keep the documentation at hand and look at it
  - Sharma, A. (2020). *Practical Apache Lucene 8*. Apress Media, New York, USA.
- Built on Lucene - Industry-grade
  - Apache Solr (https://solr.apache.org/)
  - Elasticsearch (https://www.elastic.co/)

pyLucene

https://lucene.apache.org/pylucene/



APACHE LUCENE

PyLucene | News | JCC | Issue Tracker | Mailing Lists | Lucene TLP

Welcome to PyLucene

DOWNLOAD

What is PyLucene ?

PyLucene is a Python extension for accessing Java Lucene™. Its goal is to allow you to use Lucene's text indexing and searching capabilities from Python. It is API compatible with Java Lucene version 9.7.0 as of July 12th, 2023.

PyLucene is not a Lucene port but a Python wrapper around Java Lucene. PyLucene embeds a Java VM with Lucene into a Python process. The PyLucene Python extension, a Python module called **lucene** is machine-generated by JCC.

PyLucene is built with JCC, a C++ code generator that makes it possible to call into Java classes from Python via Java's Native Invocation Interface (JNI). Sources for JCC are included with the PyLucene sources.

See here for more information and documentation about PyLucene.

Requirements

PyLucene is supported on macOS, Linux, Solaris and Windows.

PyLucene requires Python 3.x (x >= 3), and Java x (x >= 8 for PyLucene 8 and earlier, x >= 11 for PyLucene 9). Python 2 support is untested since PyLucene 9.4.1 / JCC 3.13. Building PyLucene requires GNU Make and a C++ compiler. Use of setuptools is recommended.

See the JCC installation instructions for more information about building JCC from sources.

See the PyLucene installation instructions for more information about building PyLucene from sources.

Latest News

12 July 2023 - PyLucene 9.7.0 available

This release tracks Lucene Core's 9.7.0 release and requires Java 11.
See PyLucene 9.7.0 CHANGES and JCC 3.13 CHANGES for details.
Source distributions are available here.

3 June 2023 - PyLucene 9.6.0 available

This release tracks Lucene Core's 9.6.0 release and requires Java 11.
See PyLucene 9.6.0 CHANGES and JCC 3.13 CHANGES for details.

Documentation

License
Features
Install

Events

ASF links

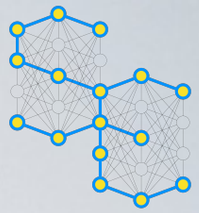
Apache Software Foundation
Thanks
Become a Sponsor
Security

Related Projects

Apache Solr
Apache Hadoop
Apache ManifoldCF
Apache Lucene.Net
Apache Mahout
Apache Nutch
Apache OpenNLP
Apache Tika
Apache Zookeeper

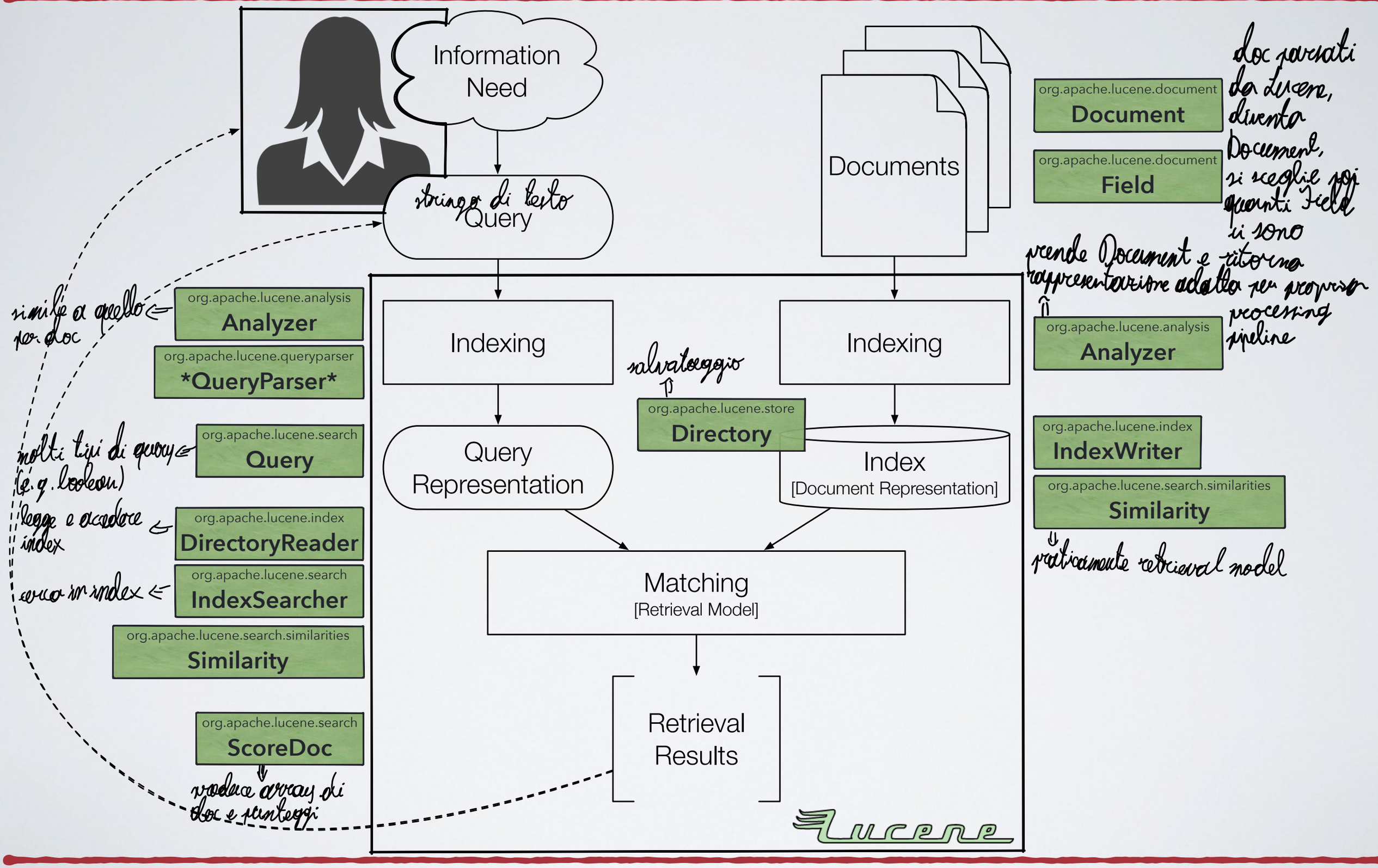
- PyLucene is not a Lucene port but a **Python wrapper around Java Lucene**
- PyLucene embeds a Java VM with Lucene into a Python process
- The PyLucene Python extension, a Python module called **lucene** is machine-generated by JCC
  - JCC is a C++ code generator that makes it possible to call into Java classes from Python via Java's Native Invocation Interface (JNI)
- More information and examples at: https://lucene.apache.org/pylucene/features.html

Alternatives to Lucene (Mostly Academic)

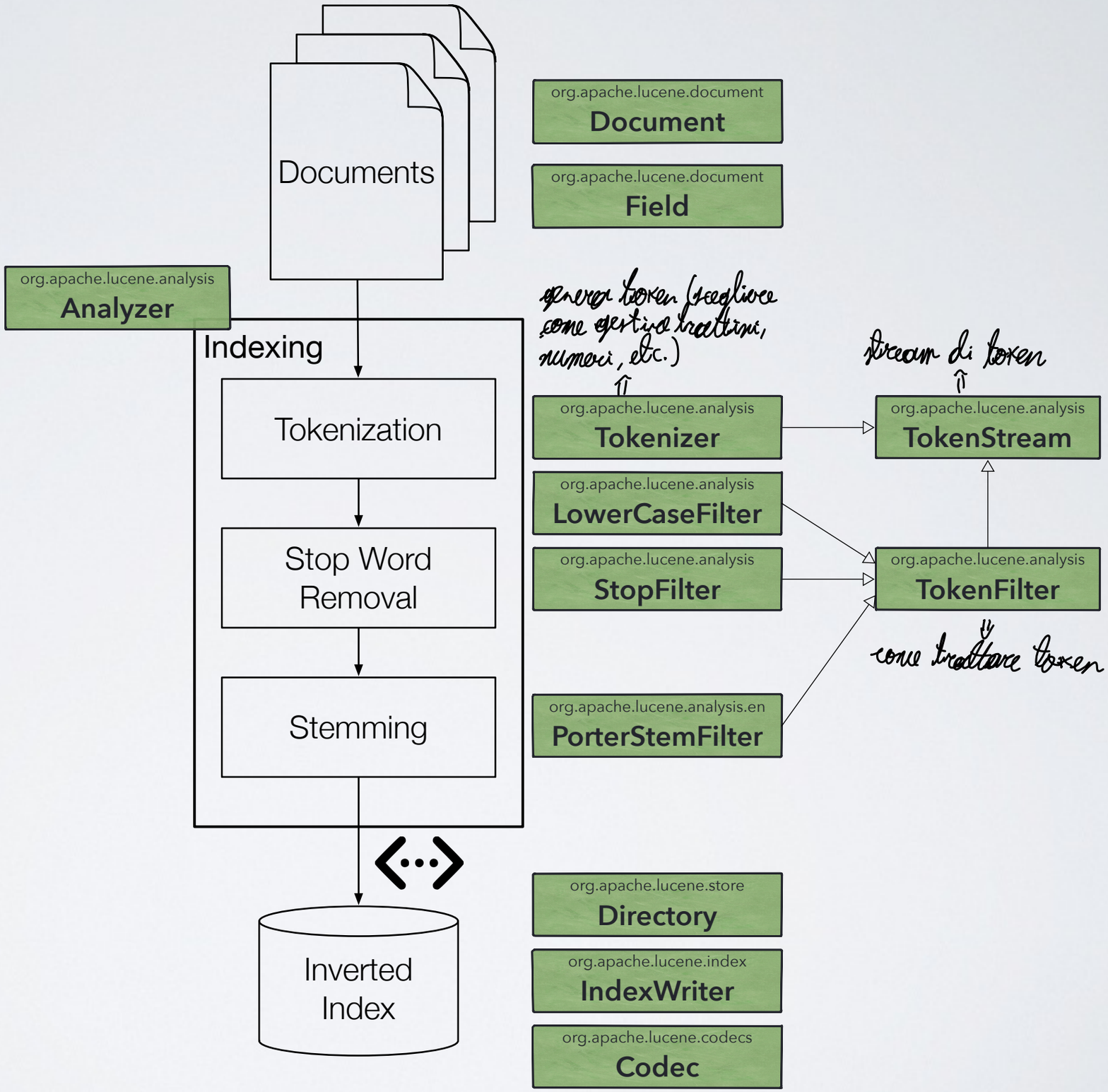


- Java
  - Terrier (http://terrier.org/) - University of Glasgow, UK;
    - also Python wrapper (https://github.com/terrier-org/pyterrier)
  - Galago (https://sourceforge.net/p/lemur/wiki/Galago/) - University of Massachusetts Amherst, USA
    - A version of Galago is used in the textbook by Croft et al.
  - MG4J (https://mg4j.di.unimi.it/) - University of Milan, Italy
- C++
  - JASSv2 (https://github.com/andrewtrotman/JASSv2), still in (not very active) development - University of Otago, New Zealand
    - also wrapped in Python
  - XAPIAN (https://xapian.org/)
- Python
  - Whoosh (https://github.com/mchaput/whoosh), not actively maintained since 5+ years
- No more supported
  - Indri (https://sourceforge.net/projects/lemur/files/lemur/indri-5.18/), written in C++ - University of Massachusetts Amherst, USA
  - ATIRE (https://github.com/andrewtrotman/ATIRE), written in C/C++ - University of Otago, New Zealand
- Built on Lucene (mostly for running IR experiments)
  - Anserini (https://github.com/castorini/anserini) - University of Waterloo, Canada
    - also Python wrapper (https://github.com/castorini/pyserini)
  - Lucindri (https://github.com/lemurproject/Lucindri) - University of Massachusetts Amherst, USA
  - GoPAL (https://bitbucket.org/frrncl/gopal/) - University of Padua, Italy

Apache Lucene: The Y



Apache Lucene: (Basic) Indexing



Hello, IR!

(Toy) Corpus

- `d1` – The quokka, the only member of the genus Setonix, is a small marsupial about the size of a domestic cat.
- `d2` – Wombats are small, short-legged, muscular quadrupedal marsupials that are native to Australia.
- `d3` – Quokkas have little fear of humans and commonly approach people closely, particularly on Rottnest Island in Australia, where a prevalent population exists.

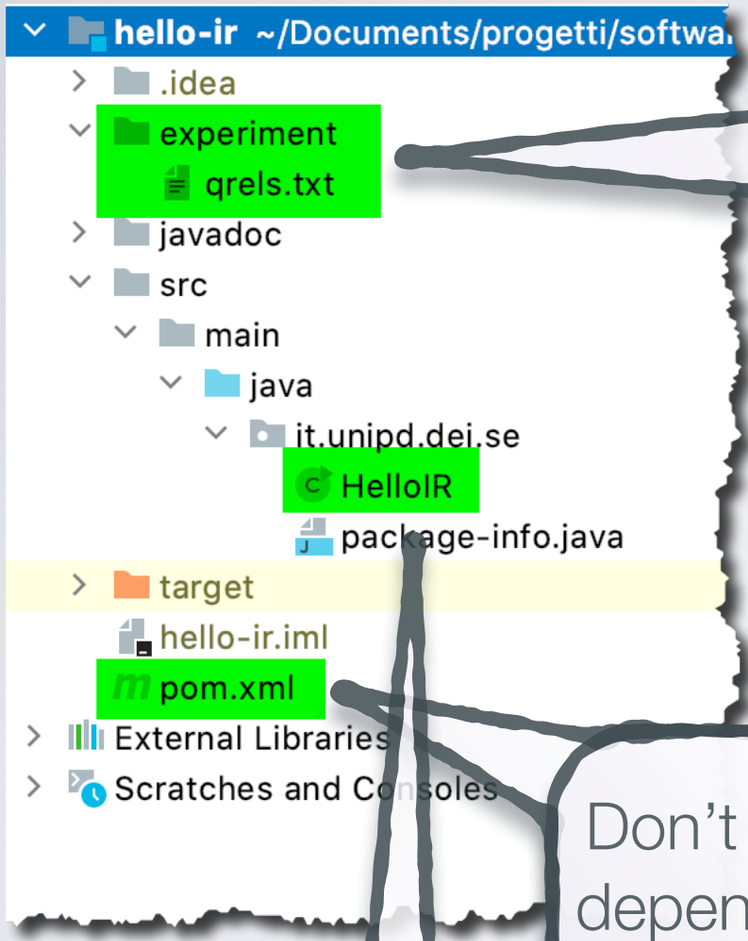
[credits to Maria Maistro]

Topics and Relevance Judgements

- 001 – quokka
- 002 – Australia animals
- 003 – small marsupial
- 004 – quokka australia

Topic ID	Fixed	Document ID	Judgement
001	0	d1	1
001	0	d2	0
001	0	d3	1
002	0	d1	1
002	0	d2	1
002	0	d3	1
003	0	d1	1
003	0	d2	1
003	0	d3	1
004	0	d1	1
004	0	d2	0
004	0	d3	1

Project Structure



The `experiment` folder will contain the Lucene index and the run.

`qrels.txt` are the relevance judgments shown before

```
    and other properties
-->
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <lucene.version>8.8.1</lucene.version>
    <java.version>15</java.version>
</properties>
```

Don't forget to add the dependencies on Lucene in the `pom.xml` file

```
<!-- Dependencies -->
<dependencies>
    <dependency>
        <groupId>org.apache.lucene</groupId>
        <artifactId>lucene-core</artifactId>
        <version>${lucene.version}</version>
    </dependency>

    <dependency>
        <groupId>org.apache.lucene</groupId>
        <artifactId>lucene-queryparser</artifactId>
        <version>${lucene.version}</version>
    </dependency>
</dependencies>
```

`HelloIR` indexes and searches the toy documents

Parsing the (toy) Documents

```java
public List<Document> parseDocuments() {

    System.out.printf("%n------------ PARSING DOCUMENTS ------------%n");

    // the list of documents
    final ArrayList<Document> docs = new ArrayList<>();

    Document d = null;

    // create a new document for D1
    d = new Document();
    d.add(new Field(ID,  value: "d1", TextField.TYPE_STORED));
    d.add(new Field(BODY, D1, TextField.TYPE_STORED));
    docs.add(d);

    // create a new document for D2
    d = new Document();
    d.add(new Field(ID,  value: "d2", TextField.TYPE_STORED));
    d.add(new Field(BODY, D2, TextField.TYPE_STORED));
    docs.add(d);

    // create a new document for D3
    d = new Document();
    d.add(new Field(ID,  value: "d3", TextField.TYPE_STORED));
    d.add(new Field(BODY, D3, TextField.TYPE_STORED));
    docs.add(d);

    System.out.printf("The documents are:%n");
    docs.forEach(System.out::println);
    System.out.printf("-------------------------------------------%n");

    return docs;
}
```

The toy documents are hard-coded as static fields of the class.

For each toy document, create a `Document` object with two `Fields`, one for storing the identifier of the toy document and the other its body.

Note that these `Fields` are of `TYPE_STORED` which means that, besides the index information, the whole content of the `Fields` is added to the index.

Indexing the (toy) Documents

```java
public void index(final List<Document> docs, final Analyzer analyzer) throws IOException {

    System.out.printf("%n------------ INDEXING DOCUMENTS ------------%n");

    // Open the directory in Lucene
    final Directory directory = FSDirectory.open(indexPath);

    // Utility class for holding all the required configuration for the indexer
    final IndexWriterConfig config = new IndexWriterConfig(analyzer);

    // force to re-create the index if it already exists
    config.setOpenMode(IndexWriterConfig.OpenMode.CREATE);

    // set the similarity. BM25 is already the default one
    config.setSimilarity(new BM25Similarity());

    // The actual indexer
    final IndexWriter writer = new IndexWriter(directory, config);

    System.out.printf("- Indexer successfully created%n");

    for (Document d : docs) {
        writer.addDocument(d);

        System.out.printf("- Document %s successfully indexed%n", d.get(ID));
    }

    writer.close();
    directory.close();
    System.out.printf("- Indexer successfully closed%n");

    System.out.printf("-------------------------------------------%n");

}
```

The **`Analyzer`** must be the same for both indexing and searching, so it is passed as a parameter.

First, open the **`Directory`** where to write te index. **`open`** picks the most efficient implementation for a platform.

Then, set the configuration for the indexer

Finally, create an **`IndexWriter`** and use it for document processing and index creation

Don't forget to **`close`** and release all the resources.

Searching the (toy) Documents

```java
public void search(final String queryID, final String query, final Analyzer analyzer) throws IOException {

    System.out.printf("%n------------ SEARCHING DOCUMENTS FOR QUERY %s ------------%n", queryID);

    // Open the directory in Lucene
    final Directory directory = FSDirectory.open(indexPath);

    // Reads the index
    final DirectoryReader reader = DirectoryReader.open(directory);

    // Searches the index
    final IndexSearcher searcher = new IndexSearcher(reader);

    // set the similarity. BM25 is already the default one
    searcher.setSimilarity(new BM25Similarity());

    System.out.printf("- Searcher successfully created%n");

    // Parses the textual query
    final SimpleQueryParser qp = new SimpleQueryParser(analyzer, BODY);

    final Query q = qp.parse(query);

    System.out.printf("- Query successfully parsed: %s%n", q.toString());

    // Perform the actual search
    final ScoreDoc[] hits = searcher.search(q, MAX_DOCS_RETRIEVED).scoreDocs;

    System.out.printf("- %d documents retrieved%n", hits.length);

    // write the list to the run file and to the console
    for (int i = 0, n = hits.length; i < n; i++) {
        String docID = reader.document(hits[i].doc).get(ID);

        // write to the run file
        run.printf(Locale.ENGLISH, format: "%s\tQ0\t%s\t%d\t%.6f\t%s%n", queryID, docID, i, hits[i].score, RUN_ID);

        // write to the console
        System.out.printf(Locale.ENGLISH, format: "   %s\tQ0\t%s\t%d\t%.6f\t%s%n", queryID, docID, i, hits[i].score,
                          RUN_ID);
    }

    // ensure the run is flushed to disk
    run.flush();

    reader.close();
    directory.close();
    System.out.printf("- Searcher successfully closed%n");

    System.out.printf("-----------------------------------------------------------%n");
}
```

Use the same **`Analyzer`** used for indexing

Open the **`Directory`**, use DirectoryReader to read the index and IndexSearcher for searching it. You must use the same **`Similarity`** used for indexing.

Parse each query

Search for the documents. The results list is the **`ScoreDoc`** array.

Write the run file using the **`trec_eval`** format

Don’t forget to **`close`** and release all the resources.

Putting Everything Together

```java
public static void main(String[] args) throws IOException {

    final HelloIR hir;

    if (args.length == 2) {
        hir = new HelloIR(args[0], args[1]);
    } else {
        hir = new HelloIR();
    }

    // create the list of documents from the provided toy ones
    final List<Document> docs = hir.parseDocuments();

    // The analyzer to be used for document pre-processing
    // The StandardAnalyzer tokenizes documents using spaces and recognizing, e.g., urls; then it turns tokens
    // lower-case and removes stop words.
    final Analyzer analyzer = new StandardAnalyzer();

    // index the documents
    hir.index(docs, analyzer);

    // search the documents
    hir.search( queryID: "001",  query: "quokka", analyzer);
    hir.search( queryID: "002",  query: "Australia animals", analyzer);
    hir.search( queryID: "003",  query: "small marsupial", analyzer);
    hir.search( queryID: "004",  query: "quokka australia", analyzer);

    // release resources
    hir.close();

}
```

Either use index and run file location provided from the command line or the default ones

Parse the documents

Pick an `Analyzer` for indexing and searching

Index the documents

Search for queries and write the run file

Close everything

Running

```
------------ INITIALIZING ------------
- Index stored at: /Users/ferro/Documents/progetti/software/search-engines/se-
- Run file at: /Users/ferro/Documents/progetti/software/search-engines/se-unipd
----------------------------------------

------------ PARSING DOCUMENTS ------------
The documents are:
Document<stored,indexed,tokenized<id:d1> stored,indexed,tokenized<body:The quok
Document<stored,indexed,tokenized<id:d2> stored,indexed,tokenized<body:Wombats
Document<stored,indexed,tokenized<id:d3> stored,indexed,tokenized<body:Quokkas
-------------------------------------------

------------ INDEXING DOCUMENTS ------------
- Indexer successfully created
- Document d1 successfully indexed
- Document d2 successfully indexed
- Document d3 successfully indexed
- Indexer successfully closed
-------------------------------------------

------------ SEARCHING DOCUMENTS FOR QUERY 001 ------------
- Searcher successfully created
- Query successfully parsed: body:quokka
- 1 documents retrieved
   001  Q0  d1  0   0.429845    hello-IR
- Searcher successfully closed
------------------------------------------------------------

------------ SEARCHING DOCUMENTS FOR QUERY 002 ------------
- Searcher successfully created
- Query successfully parsed: body:australia body:animals
- 2 documents retrieved
   002  Q0  d2  0   0.242497    hello-IR
   002  Q0  d3  1   0.197481    hello-IR
- Searcher successfully closed
------------------------------------------------------------
```

Compiling trec_eval

```
√ trec_eval-9.0.7 % make
gcc -g -I.  -Wall -DVERSIONID=\"9.0.7\"  -o trec_eval trec_eval.c formats.c meas_init.c meas_acc.c meas_avg.c meas_print_single.c meas_print_
final.c get_qrels.c get_trec_results.c get_prefs.c get_qrels_prefs.c get_qrels_jg.c form_res_rels.c form_res_rels_jg.c form_prefs_counts.c ut
ility_pool.c get_zscores.c convert_zscores.c measures.c  m_map.c m_P.c m_num_q.c m_num_ret.c m_num_rel.c m_num_rel_ret.c m_gm_map.c m_Rprec.c
 m_recip_rank.c m_bpref.c m_iprec_at_recall.c m_recall.c m_Rprec_mult.c m_utility.c m_11pt_avg.c m_ndcg.c m_ndcg_cut.c m_Rndcg.c m_ndcg_rel.c
 m_binG.c m_G.c m_rel_P.c m_success.c m_infap.c m_map_cut.c m_gm_bpref.c m_runid.c m_relstring.c m_set_P.c m_set_recall.c m_set_rel_P.c m_set
_map.c m_set_F.c m_num_nonrel_judged_ret.c m_prefs_num_prefs_poss.c m_prefs_num_prefs_ful.c m_prefs_num_prefs_ful_ret.c m_prefs_simp.c m_pref
s_pair.c m_prefs_avgjg.c m_prefs_avgjg_Rnonrel.c m_prefs_simp_ret.c m_prefs_pair_ret.c m_prefs_avgjg_ret.c m_prefs_avgjg_Rnonrel_ret.c m_pref
s_simp_imp.c m_prefs_pair_imp.c m_prefs_avgjg_imp.c m_map_avgjg.c m_Rprec_mult_avgjg.c m_P_avgjg.c m_yaap.c -lm
√ trec_eval-9.0.7 %
```

- Go into the source directory and compile `trec_eval` by running **`make`** (warnings may arise)
- Once done, you should see an executable named **`trec_eval`**

```
-rw-rw-r--@  1 ferro  staff    1476 10 Ott  2019 sysfunc.h
drwxrwxr-x@ 22 ferro  staff     704 10 Ott  2019 test
-rwxr-xr-x   1 ferro  staff  206684  7 Mar 18:34 trec_eval
-rw-rw-r--@  1 ferro  staff   19680 10 Ott  2019 trec_eval.c
drwxr-xr-x@  3 ferro  staff      96  6 Mar 17:01 trec_eval.dSYM
-rw-rw-r--@  1 ferro  staff    9520 10 Ott  2019 trec_eval.h
-rw-rw-r--@  1 ferro  staff   11479 10 Ott  2019 trec_format.h
-rw-rw-r--@  1 ferro  staff    1652 10 Ott  2019 utility_pool.c
√ trec_eval-9.0.7 %
```

Getting Help from trec_eval

```
[√ trec_eval-9.0.7 % ./trec_eval -h
trec_eval [-h] [-q] [-m measure[.params] [-c] [-n] [-l <num>]
   [-D debug_level] [-N <num>] [-M <num>] [-R rel_format] [-T results_format]
   rel_info_file  results_file

Calculate and print various evaluation measures, evaluating the results
in results_file against the relevance info in rel_info_file.

There are a fair number of options, of which only the lower case options are
normally ever used.
 --help:
 -h: Print full help message and exit. Full help message will include
     descriptions for any measures designated by a '-m' parameter, and
     input file format descriptions for any rel_info_format given by '-R'
     and any top results_format given by '-T.'
     Thus to see all info about preference measures use
          trec_eval -h -m all_prefs -R prefs -T trec_results
--version:
-v: Print version of trec_eval and exit.
--query_eval_wanted:
-q: In addition to summary evaluation, give evaluation for each query/topic
--measure measure_name[.measure_params]:
-m measure: Add 'measure' to the lists of measures to calculate and print.
   If 'measure' contains a '.', then the name of the measure is everything
   preceeding the period, and everything to the right of the period is
   assumed to be a list of parameters for the measure, separated by ','.
   There can be multiple occurrences of the -m flag.
   'measure' can also be a nickname for a set of measures. Current
   nicknames include
      'official': the main measures often used by TREC
      'all_trec': all measures calculated with the standard TREC
                  results and rel_info format files.
      'set': subset of all_trec that calculates unranked values.
      'prefs': Measures not in all_trec that calculate preference measures.
```

- To get help run `trec_eval -h`

Using trec_eval to Compute Set-based Measures

- To compute set-based evaluation measures (Precision, Recall, …) run

  `trec_eval -q -m set qrels.txt run.txt`

  - `-q` prints topic-by-topic results
  - `-m` selects which measures to compute, use `set` for set-based evaluation measures

```
[√ trec_eval-9.0.7 % ./trec_eval -q -m set ../hello-ir/experiment/qrels.txt ../hello-ir/experiment/run.txt
num_ret                 001     1
num_rel                 001     2
num_rel_ret             001     1
utility                 001     1.0000
set_P                   001     1.0000
set_relative_P          001     1.0000
set_recall              001     0.5000
set_map                 001     0.5000
set_F                   001     0.6667
num_ret                 002     2
num_rel                 002     3
num_rel_ret             002     2
utility                 002     2.0000
set_P                   002     1.0000
set_relative_P          002     1.0000
set_recall              002     0.6667
set_map                 002     0.6667
set_F                   002     0.8000
num_ret                 003     2
```

`num_ret` is the total number of retrieved documents

`num_rel` is the total number of relevant documents for that topic

`num_rel_ret` is the total number of relevant retrieved documents

`set_P` is set-based Precision

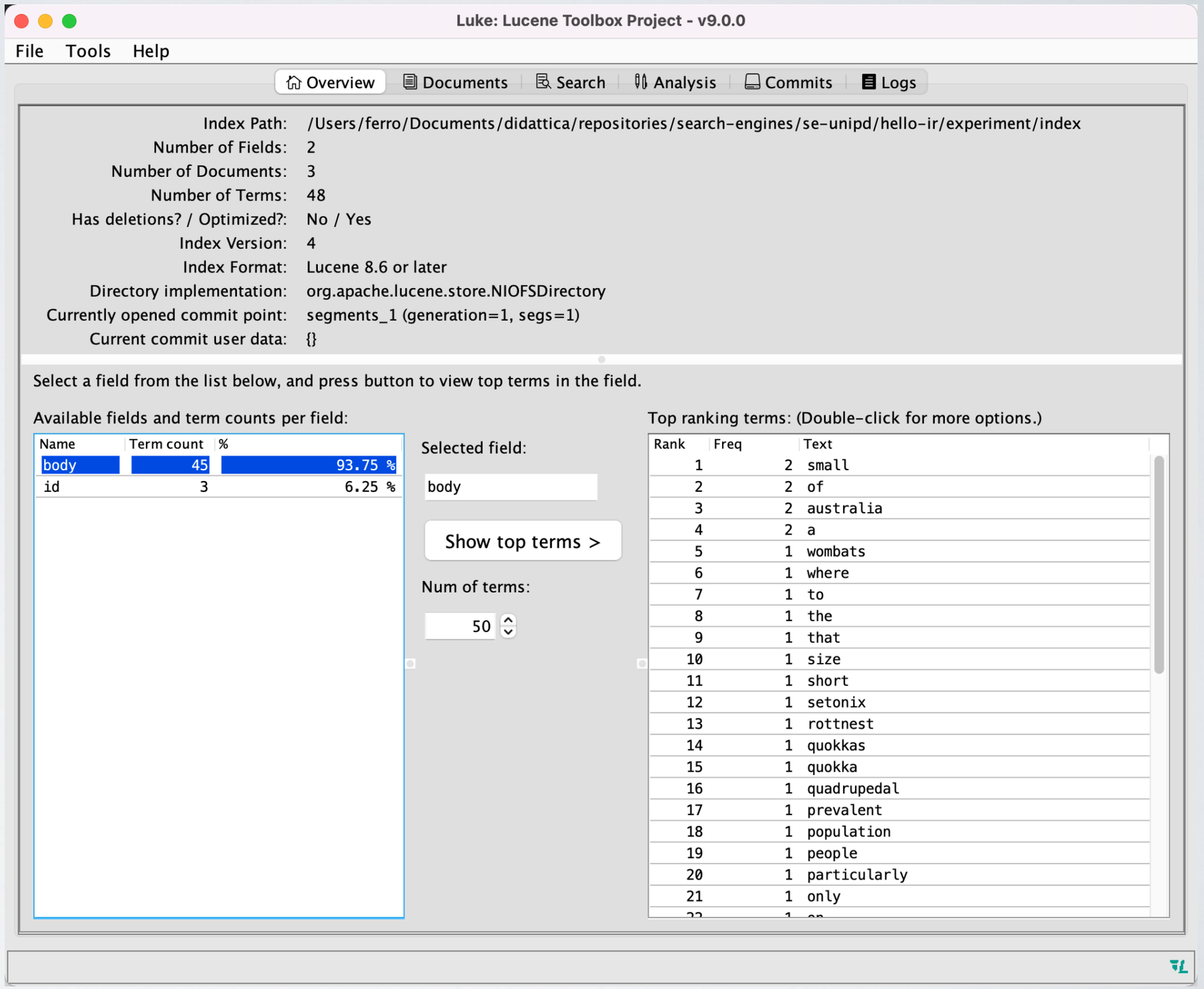
`set_recall` is set-based Recall

Will your Homework Be That Different?

- You will develop and implement an IR system
  - Possibly not as simple as `HelloIR`
- You will index the corpus provided by the CLEF organizers
- You will search for the topics provided by the CLEF organizers
- You will produce one (or more) runs (`seupd2324-<team acronym>-<run info>.txt`)
- After a while, CLEF organizers will give you back relevance judgements (`qrels.txt`)
  - In the meantime, you will write the report for homework 1
- You will use the provided relevance judgement to score the performance of your runs
  - You will update the homework 1 report with a discussion and analysis of the performance of your system. This will be the paper submitted to CLEF
  - You will give a talk about the whole story

Lucene Luke: Inspecting the Index

- In the Lucene binary distribution you will find a folder called `luke`
- Use either `luke.sh` or `luke.bat` to start luke



Character Encoding

The ASCII Character Encoding

- **ASCII** (American Standard Code for Information Interchange) is a character encoding scheme introduced in 1963 by the American Standards Association
- It uses **7 bits** to represents **128 characters** — control characters, latin alphabet letters (lower and upper cases), numbers, punctuation, some symbols
- It has been then standardised by ISO in 1972. Since ASCII did not provide a number of characters needed in languages other than English, a number of national variants were made that substituted a few less-used characters with needed ones, leading to incompatibilities

ASA (1963). *American Standard Code for Information Interchange – X3.4- 1963*. American Standards Association (ASA), USA.

ISO/IEC 646 (1972). *Information processing – ISO 7-bit coded character set for information interchange*. Recommendation ISO/IEC 646:1972.

X.3.4-1963: Cover and Code Table

ASA
Reg. U. S. Pat. Off.
X3.4-1963
UDC 681.3 (Tentative)

→ Superceded by X3.4-1965

American Standard Code for
Information Interchange

Sponsor

Business Equipment Manufacturers Association

Approved June 17, 1963

AMERICAN STANDARDS ASSOCIATION
INCORPORATED

American Standard Code for

Information Interchange

1. Scope

This coded character set is to be used for the general interchange of information among information processing systems, communication systems, and associated equipment.

2. Standard Code

b_7				0	0	0	0	1	1	1	1
b_6				0	0	1	1	0	0	1	1
b_5				0	1	0	1	0	1	0	1
b_4	b_3	b_2	b_1								
0	0	0	0	NULL	DC_0	b	0	@	P		
0	0	0	1	SOM	DC_1	!	1	A	Q		
0	0	1	0	EOA	DC_2	"	2	B	R		
0	0	1	1	EOM	DC_3	#	3	C	S		
0	1	0	0	EOT	DC_4 (STOP)	$	4	D	T		
0	1	0	1	WRU	ERR	%	5	E	U		
0	1	1	0	RU	SYNC	&	6	F	V		
0	1	1	1	BELL	LEM	' (APOS)	7	G	W		
1	0	0	0	FE_0	S_0	(	8	H	X		
1	0	0	1	HT / SK	S_1	)	9	I	Y		
1	0	1	0	LF	S_2	*	:	J	Z		
1	0	1	1	V_{TAB}	S_3	+	;	K	[		
1	1	0	0	FF	S_4	, (COMMA)	<	L	\		
1	1	0	1	CR	S_5	−	=	M	]		
1	1	1	0	SO	S_6	.	>	N	↑		
1	1	1	1	SI	S_7	/	?	O	←		

UNASSIGNED

ACK
(1)
ESC
DEL

UNASSIGNED

3. Positional Order and Notation

Standard 7-bit set code positional order and notation are shown below with b_7 the high-order, and b_1 the low-order, bit position.

	b_7	b_6	b_5	b_4	b_3	b_2	b_1
EXAMPLE: The code for "R" is:	1	0	1	0	0	1	0

4. Legend

NULL	Null/Idle	DC_1-DC_3	Device control
SOM	Start of message	DC_4(Stop)	Device control (stop)
EOA	End of address	ERR	Error

Legend continued on following page

Extended ASCII

- The ASCII encoding has been extended to include also non-English symbols to, e.g., have a better coverage of European languages
- The so-called "extended ASCII" uses **8 bits** to encode **256 characters**.
  - the first 128 characters are the same as in ASCII at 7 bits
  - the additional (upper) 128 characters are used to define a set of alternative code tables, e.g. for different European and non-European languages, leading to several compatibility issues
- Extended ASCII is standardised in the ISO 8859 sets of recommendations since 1987

ISO 8859 (1987). *Information processing – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*. Recommendation ISO 8859-1:1987.

The Unicode Standard

- In 1991 the Unicode Consortium (`https://home.unicode.org/`) developed a new standard to address the compatibility issues among the different ASCII encodings and to develop a single set of characters suitable for all the different alphabets and symbols
- The first versions of Unicode used **16 bits** to represent **65,536 characters** while the more recent versions use **32 bits** to represent up to **4,294,967,296** possible **characters**
  - the first 256 characters are in common with the ISO 8859-1 standard
- To save memory, alternative encoding schemes have developed for “packing” Unicode symbols, called **Unicode Transformation Format (UTF)**
  - **UTF-8** is among the most adopted: it uses 8 bits for the characters which are in common with extended ASCII, 16 bits for the new characters added by the first Unicode versions, and 32 bits only when needed to represents the newest characters
- It has been standardised by ISO in 1993 as **Universal Character Set (UCS)**

ISO 10646 (1993). *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane*. Recommendation ISO/IEC 10646-1:1993.

Example of Extended ASCII and Unicode

ASCII/8859-1 Text

A	0100 0001
S	0101 0011
C	0100 0011
I	0100 1001
I	0100 1001
/	0010 1111
8	0011 1000
8	0011 1000
5	0011 0101
9	0011 1001
-	0010 1101
1	0011 0001
	0010 0000
t	0111 0100
e	0110 0101
x	0111 1000
t	0111 0100

Unicode Text

A	0000 0000 0100 0001
S	0000 0000 0101 0011
C	0000 0000 0100 0011
I	0000 0000 0100 1001
I	0000 0000 0100 1001
	0000 0000 0010 0000
天	0101 1001 0010 1001
地	0101 0111 0011 0000
	0000 0000 0010 0000
س	0000 0110 0011 0011
ل	0000 0110 0100 0100
ا	0000 0110 0010 0111
م	0000 0110 0100 0101
	0000 0000 0010 0000
α	0000 0011 1011 0001
≴	0010 0010 0111 0000
γ	0000 0011 1011 0011

Example of Unicode Tables

C0 Controls and Basic Latin

	000	001	002	003	004	005	006	007
0	NUL 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070
1	SOH 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071
2	STX 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072
3	ETX 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073
4	EOT 0004	DC4 0014	$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074
5	ENQ 0005	NAK 0015	% 0025	5 0035	E 0045	U 0055	e 0065	u 0075
6	ACK 0006	SYN 0016	& 0026	6 0036	F 0046	V 0056	f 0066	v 0076
7	BEL 0007	ETB 0017	' 0027	7 0037	G 0047	W 0057	g 0067	w 0077
8	BS 0008	CAN 0018	(0028	8 0038	H 0048	X 0058	h 0068	x 0078
9	HT 0009	EM 0019	) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079
A	LF 000A	SUB 001A	* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A
B	VT 000B	ESC 001B	+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B
C	FF 000C	FS 001C	, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C
D	CR 000D	GS 001D	- 002D	= 003D	M 004D	] 005D	m 006D	} 007D
E	SO 000E	RS 001E	. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E
F	SI 000F	US 001F	/ 002F	? 003F	O 004F	_ 005F	o 006F	DEL 007F

Greek and Coptic

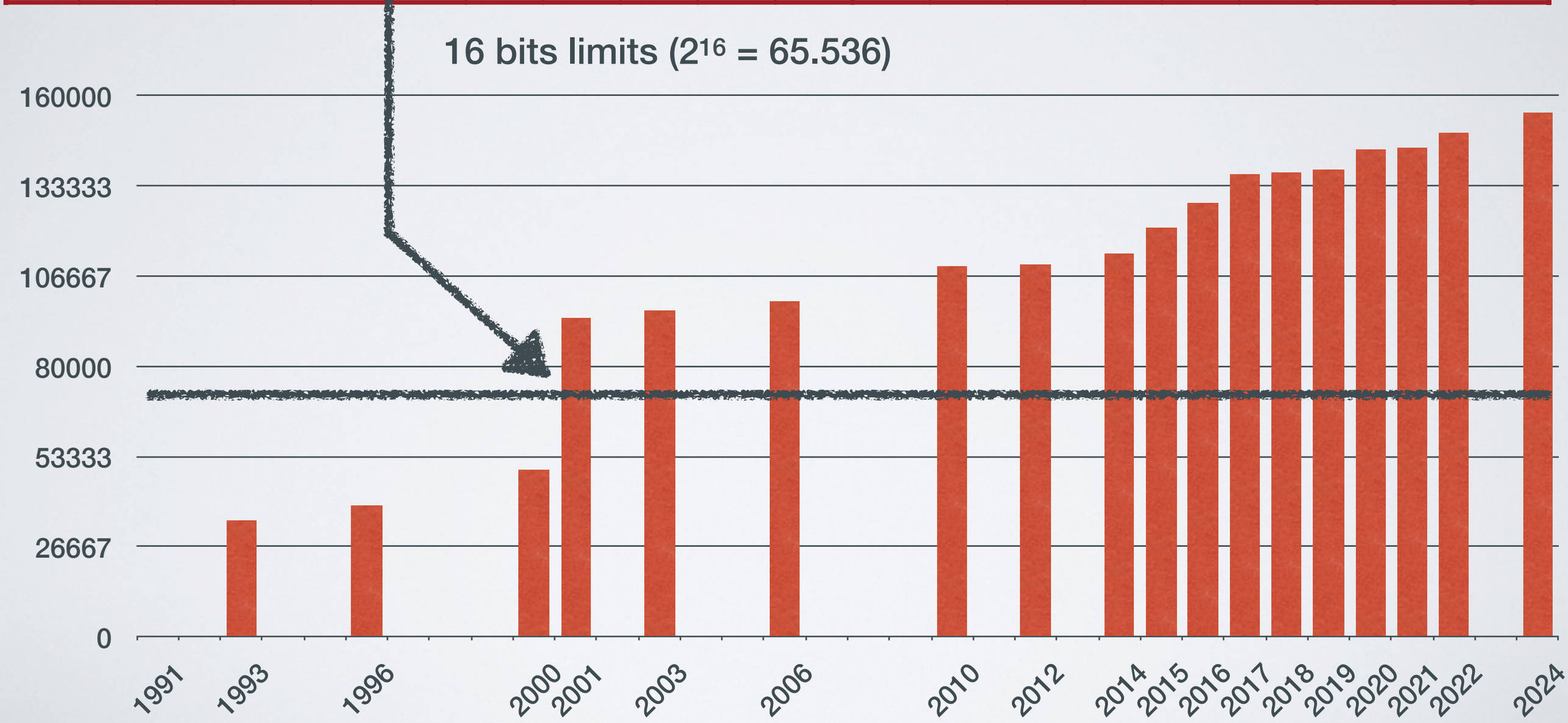
	037	038	039	03A	03B	03C	03D	03E	03F
0	Ͱ 0370		ΐ 0390	Π 03A0	ΰ 03B0	π 03C0	ϐ 03D0	Ϣ 03E0	ϰ 03F0
1	ͱ 0371		Α 0391	Ρ 03A1	α 03B1	ρ 03C1	ϑ 03D1	ϣ 03E1	ϱ 03F1
2	Ͳ 0372		Β 0392		β 03B2	ς 03C2	ϒ 03D2	Ϥ 03E2	ϲ 03F2
3	ͳ 0373		Γ 0393	Σ 03A3	γ 03B3	σ 03C3	ϓ 03D3	ϥ 03E3	ϳ 03F3
4	ʹ 0374	΄ 0384	Δ 0394	Τ 03A4	δ 03B4	τ 03C4	ϔ 03D4	Ϧ 03E4	ϴ 03F4
5	͵ 0375	΅ 0385	Ε 0395	Υ 03A5	ε 03B5	υ 03C5	ϕ 03D5	ϧ 03E5	ϵ 03F5
6	Ͷ 0376	Ά 0386	Ζ 0396	Φ 03A6	ζ 03B6	φ 03C6	ϖ 03D6	Ϩ 03E6	϶ 03F6
7	ͷ 0377	· 0387	Η 0397	Χ 03A7	η 03B7	χ 03C7	ϗ 03D7	ϩ 03E7	Ϸ 03F7
8		Έ 0388	Θ 0398	Ψ 03A8	θ 03B8	ψ 03C8	Ϙ 03D8	Ϫ 03E8	ϸ 03F8
9		Ή 0389	Ι 0399	Ω 03A9	ι 03B9	ω 03C9	ϙ 03D9	ϫ 03E9	Ϲ 03F9
A	ͺ 037A	Ί 038A	Κ 039A	Ϊ 03AA	κ 03BA	ϊ 03CA	Ϛ 03DA	Ϭ 03EA	Ϻ 03FA
B	ͻ 037B		Λ 039B	Ϋ 03AB	λ 03BB	ϋ 03CB	ϛ 03DB	ϭ 03EB	ϻ 03FB
C	ͼ 037C	Ό 038C	Μ 039C	ά 03AC	μ 03BC	ό 03CC	Ϝ 03DC	Ϯ 03EC	ϼ 03FC
D	ͽ 037D		Ν 039D	έ 03AD	ν 03BD	ύ 03CD	ϝ 03DD	ϭ 03ED	Ͻ 03FD
E	; 037E	Ύ 038E	Ξ 039E	ή 03AE	ξ 03BE	ώ 03CE	Ϟ 03DE	Ϯ 03EE	Ͼ 03FE
F		Ώ 038F	Ο 039F	ί 03AF	ο 03BF	Ϗ 03CF	ϟ 03DF	ϯ 03EF	Ͽ 03FF

0400 Cyrillic 04FF

	040	041	042	043	044	045	046	047	048	049	04A	04B	04C	04D	04E	04F
0	È 0400	А 0410	Р 0420	а 0430	р 0440	è 0450	Ѡ 0460	Ѱ 0470	Ҁ 0480	Ґ 0490	Қ 04A0	Ұ 04B0	І 04C0	Ӑ 04D0	З 04E0	Ӯ 04F0
1	Ё 0401	Б 0411	С 0421	б 0431	с 0441	ё 0451	ѡ 0461	ѱ 0471	ҁ 0481	ґ 0491	қ 04A1	ұ 04B1	Ӂ 04C1	ӑ 04D1	з 04E1	ӯ 04F1
2	Ђ 0402	В 0412	Т 0422	в 0432	т 0442	ђ 0452	Ѣ 0462	Ѳ 0472	≠ 0482	Ғ 0492	Ҋ 04A2	Ҳ 04B2	ӂ 04C2	Ӓ 04D2	Ӣ 04E2	Ӱ 04F2
3	Ѓ 0403	Г 0413	У 0423	г 0433	у 0443	ѓ 0453	ѣ 0463	ѳ 0473	◌҃ 0483	ғ 0493	ҋ 04A3	ҳ 04B3	Ҍ 04C3	ӓ 04D3	ӣ 04E3	ӱ 04F3
4	Є 0404	Д 0414	Ф 0424	д 0434	ф 0444	є 0454	Ѥ 0464	Ѵ 0474	◌҄ 0484	Ҕ 0494	Ҍ 04A4	Ҵ 04B4	ҍ 04C4	Ӕ 04D4	Ӥ 04E4	Ӵ 04F4
5	Ѕ 0405	Е 0415	Х 0425	е 0435	х 0445	ѕ 0455	ѥ 0465	ѵ 0475	◌҅ 0485	ҕ 0495	ҍ 04A5	ҵ 04B5	Ҏ 04C5	ӕ 04D5	ӥ 04E5	ӵ 04F5
6	І 0406	Ж 0416	Ц 0426	ж 0436	ц 0446	і 0456	Ѧ 0466	Ѷ 0476	◌҆ 0486	Ж 0496	Ҧ 04A6	Ҷ 04B6	ҏ 04C6	Ӗ 04D6	Ӧ 04E6	Ӷ 04F6
7	Ї 0407	З 0417	Ч 0427	з 0437	ч 0447	ї 0457	ѧ 0467	ѷ 0477	◌҇ 0487	ж 0497	ҧ 04A7	ҷ 04B7	Ґ 04C7	ӗ 04D7	ӧ 04E7	ӷ 04F7
8	Ј 0408	И 0418	Ш 0428	и 0438	ш 0448	ј 0458	Ѩ 0468	Ѹ 0478	◌҈ 0488	Ҙ 0498	Ҩ 04A8	Ҹ 04B8	ґ 04C8	Ә 04D8	Ө 04E8	Ӹ 04F8
9	Љ 0409	Й 0419	Щ 0429	й 0439	щ 0449	љ 0459	ѩ 0469	ѹ 0479	◌҉ 0489	ҙ 0499	ҩ 04A9	ҹ 04B9	Ӆ 04C9	ә 04D9	ө 04E9	ӹ 04F9
A	Њ 040A	К 041A	Ъ 042A	к 043A	ъ 044A	њ 045A	Ѫ 046A	Ѻ 047A	Ҋ 048A	Қ 049A	Ҫ 04AA	Һ 04BA	ӆ 04CA	Ӛ 04DA	Ө 04EA	Ҕ 04FA
B	Ћ 040B	Л 041B	Ы 042B	л 043B	ы 044B	ћ 045B	ѫ 046B	ѻ 047B	ҋ 048B	қ 049B	ҫ 04AB	һ 04BB	Ҷ 04CB	ӛ 04DB	ө 04EB	ҕ 04FB
C	Ќ 040C	М 041C	Ь 042C	м 043C	ь 044C	ќ 045C	Ѭ 046C	Ѽ 047C	Ҍ 048C	Ҝ 049C	Ҭ 04AC	Ҽ 04BC	ҷ 04CC	Ӝ 04DC	Ӭ 04EC	Җ 04FC
D	Ѝ 040D	Н 041D	Э 042D	н 043D	э 044D	ѝ 045D	ѭ 046D	ѽ 047D	ҍ 048D	ҝ 049D	ҭ 04AD	ҽ 04BD	Ҹ 04CD	ӝ 04DD	ӭ 04ED	җ 04FD
E	Ў 040E	О 041E	Ю 042E	о 043E	ю 044E	ў 045E	Ѯ 046E	Ѿ 047E	Ҏ 048E	Ҟ 049E	Ү 04AE	Ҿ 04BE	ҹ 04CE	Ӟ 04DE	Ӯ 04EE	Ӿ 04FE
F	Џ 040F	П 041F	Я 042F	п 043F	я 044F	џ 045F	ѯ 046F	ѿ 047F	ҏ 048F	ҟ 049F	ү 04AF	ҿ 04BF	Ӏ 04CF	ӟ 04DF	ӯ 04EF	ӿ 04FF

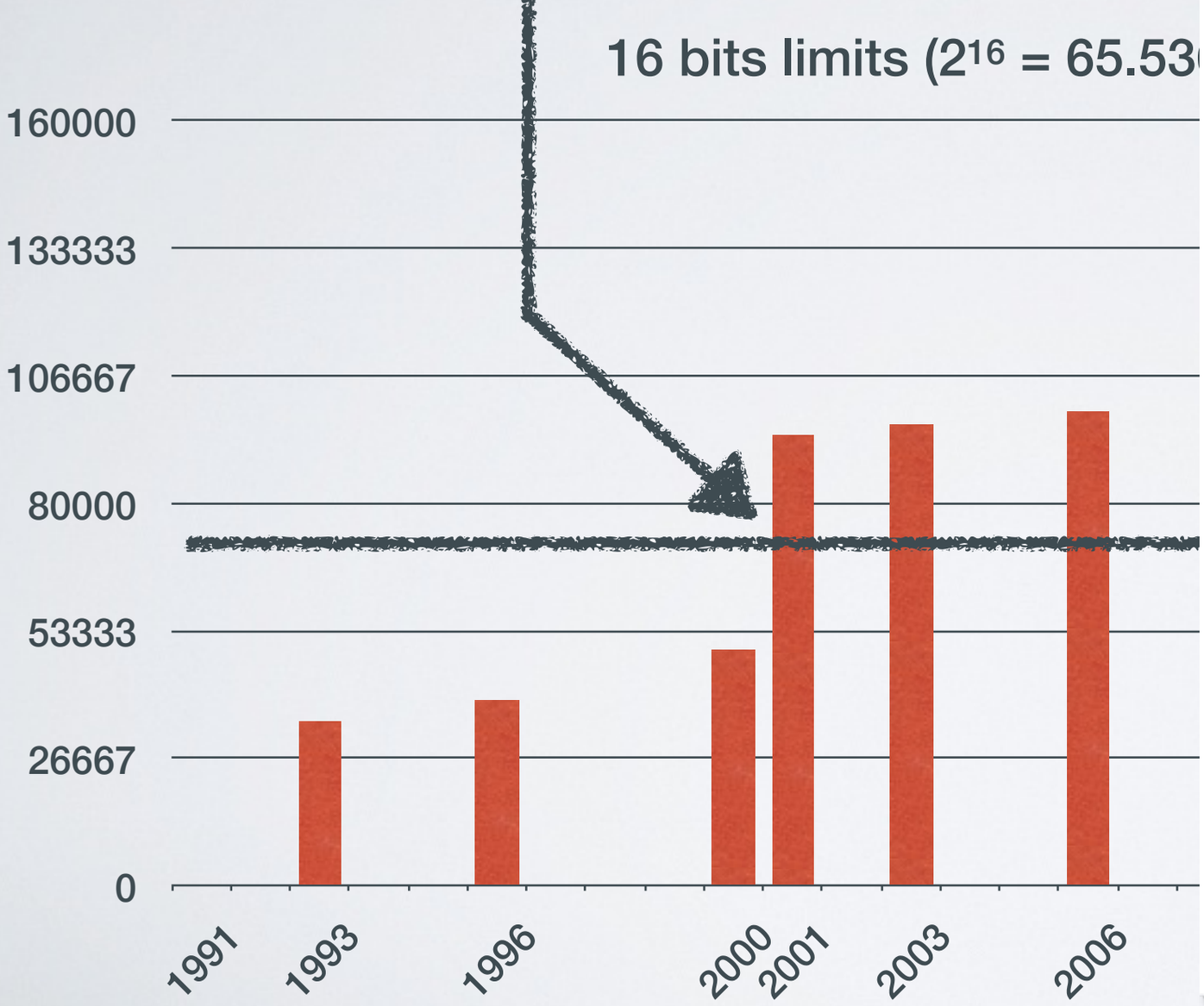
Number of Unicode Symbols

	1.0 (1991)	1.1 (1993)	2.0 (1996)	3.0 (2000)	3.1 (2001)	4.0 (2003)	5.0 (2006)	6.0 (2010)	6.1 (2012)	7.0 (2014)	8.0 (2015)	9.0 (2016)	10.0 (2017)	11.0 (2018)	12.0 (2019)	13.0 (2020)	14.0 (2021)	15.0 (2022)	16.0 (2024)
Chars	28,359	34,233	38,950	49,259	94,205	96,447	99,089	109,449	110,181	113,015	120,731	128,172	136,690	137,374	137,929	143,859	144,697	149,186	155,000



Number of Unicode Symbols

	1.0 (1991)	1.1 (1993)	2.0 (1996)	3.0 (2000)	3.1 (2001)	4.0 (2003)	5.0 (2006)	6.0 (2010)	6.1 (2012)	7.0 (2014)	6.0 (24)
Chars	28,359	34,233	38,950	49,259	94,205	96,447	99,089	109,449	110,181	113,015	00



Smileys & Emotion

face-concerned

№	Code	EPed	Sample	CLDR Short Name
1	1F971	🥱	🥱	yawning face

emotion

№	Code	EPed	Sample	CLDR Short Name
2	1F90D	🤍	🤍	white heart
3	1F90E	🤎	🤎	brown heart

People & Body

hand-fingers-partial

№	Code	EPed	Sample	CLDR Short Name
4	1F90F	🤏	🤏	pinching hand

body-parts

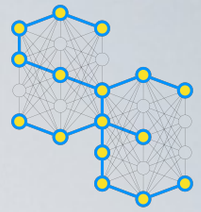
№	Code	EPed	Sample	CLDR Short Name
5	1F9BE	🦾	🦾	mechanical arm
6	1F9BF	🦿	🦿	mechanical leg
7	1F9BB	🦻	🦻	ear with hearing aid

Hello, TIPSTER!

The TIPSTER Corpus

- News articles, US government reports, … — Disks 4 and 5 excluding Congressional Record subcollection
  - LATIMES — Los Angeles Times, news articles
  - FT — Financial Times, news articles
  - FBIS — Foreign Broadcast Information Service, open source intelligence within CIA from news and media with US
  - FR94 — Federal Register, daily journal of US
- Size
  - 615 Mbytes compressed, 2 Gbytes uncompressed
  - 2,295 files
  - 528,155 documents

Documents... Do They Really Look Like That?

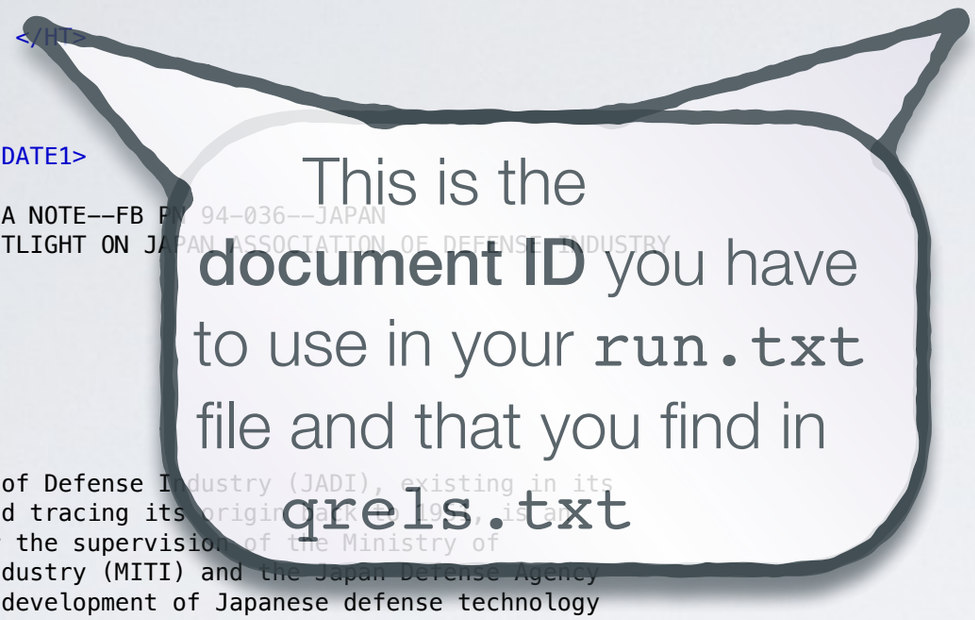


<DOC>
<DOCNO> LA010189-0001 </DOCNO>
<DOCID> 1 </DOCID>
<DATE>
<P>
January 1, 1989, Sunday, Home Edition
</P>
</DATE>
<SECTION>
<P>
Book Review; Page 1; Book Review Desk
</P>
</SECTION>
<LENGTH>
<P>
1206 words
</P>
</LENGTH>
<HEADLINE>
<P>
NEW FALLOUT FROM CHERNOBYL;
</P>
<P>
THE SOCIAL IMPACT OF THE CHERNOBYL DISASTER BY DAVID R. MARPLES (ST. MARTIN'S
PRESS: $35, CLOTH; $14.95, PAPER; 316 PP., ILLUSTRATED; 0-312-02432-0)
</P>
</HEADLINE>
<BYLINE>
<P>
By James E. Oberg , Oberg, a space engineer in Houston, is the author of
Uncovering Soviet Disasters: Exploring the Limits of Glasnost (Random House).
</P>
</BYLINE>
<TEXT>
<P>
The onset of the new Gorbachev policy of glasnost, commonly mistranslated as
openness but closer in connotation to candor or publicizing, has complicated
the task of Soviet secret-keepers and has allowed substantial new Western
insights into Soviet society. David R. Marples' new book, his second on the
Chernobyl accident of April 26, 1986, is a shining example of the best type
of
non-Soviet analysis into topics that only recently were absolutely taboo in
Moscow official circles.
</P>
<P>
The author, a British-educated historian and economist, is a research
associate
with the Canadian Institute of Ukrainian Studies at the University of
Alberta,
and the academic style of the book is undisguised. However, its intended
audience is the general public, and anyone interested in nuclear power, or
Soviet economy and society, or human drama, or just plain sleuthing state
secrets, will find hitherto unpublished revelations and explanations of the

This is the **document ID** you have to use in your `run.txt` file and that you find in `qrels.txt`

<DOC>
<DOCNO>FT911-1</DOCNO>
<PROFILE>_AN-BENBQAD8FT</PROFILE>
<DATE>910514
</DATE>
<HEADLINE>
FT 14 MAY 91 / (CORRECTED) Jubilee of a jet that did what it was designed
to do
</HEADLINE>
<TEXT>
Correction (published 16th May 1991) appended to this article.
'FRANK, it flies]' shouted someone at Sir Frank Whittle during the maiden
flight of a British jet. 'Of course it does,' replied Sir Frank, who
patented the first aircraft gas turbine. 'That's what it was bloody well
designed to do, wasn't it?'
Exactly 50 years ago yesterday, the first British jet made a brief 17-minute
flight from RAF Cranwell in Lincolnshire. To celebrate the event, Mr Eric
'Winkle' Brown, a 72-year-old test pilot of the prototype Gloster Whittle
jet, Mr Geoffrey Bone, a 73-year-old engineer, and Mr Charles McClure, a
75-year-old pilot, returned to RAF Cranwell. They are seen in front of a
restored Meteor NF 11. Sir Frank was unable to attend because of ill-health.
The Gloster Whittle was not the first jet to fly: a Heinkel 178 had its
maiden flight in August 1939, 21 months before the British aircraft.
Correction (published 16th May 1991).
THE PICTURE of a Gloster Whittle jet on Page 7 of the issue of Tuesday May
14, was taken at Bournemouth Airport and not at RAF Cranwell as stated in
the caption.
</TEXT>
<PUB>The Financial Times
</PUB>
<PAGE>
London Page 7 Photograph (Omitted).
</PAGE>
</DOC>
<DOC>
<DOCNO>FT911-2</DOCNO>
<PROFILE>_AN-BENBQABQFT</PROFILE>
<DATE>910514
</DATE>
<HEADLINE>
FT 14 MAY 91 / (CORRECTED) UK Company News: Geevor merger hits rocks over
pre-conditions
</HEADLINE>
<BYLINE>
   By KENNETH GOODING, Mining Correspondent
</BYLINE>
<TEXT>
Correction (published 16th May 1991) appended to this article.
Geevor, the UK mining group which has been fighting for survival since the
Canadian Imperial Bank of Commerce called in a Pounds 2.1m loan in
extraordinary circumstances in January, has suffered another set-back.
Its proposed merger with European Mining Finance, a Luxembourg-quoted
investment company, has run into problems and will not go ahead on the terms

Documents… Do They Really Look Like That?




```
<DOC>
<DOCNO> FBIS3-50 </DOCNO>
<HT>    "cr00000015994001" </HT>

<HEADER>
<DATE1>   23 March 1994 </DATE1>
Article Type:FBIS
Document Type:FOREIGN MEDIA NOTE--FB PY 94-036--JAPAN
<H3> <TI>      JAPAN:  SPOTLIGHT ON JAPAN ASSOCIATION OF DEFENSE INDUSTRY
</TI></H3>

</HEADER>

<TEXT>
The     Japan Association of Defense Industry (JADI), existing in its
present form since 1988 and tracing its origins back to 1952, is an
industry association under the supervision of the Ministry of
International Trade and Industry (MITI) and the Japan Defense Agency
(JDA).  JADI promotes the development of Japanese defense technology
and equipment, monitors foreign technology, lobbies on behalf of its
corporate members for government defense spending, and cooperates
with the government on export controls.

   ASSOCIATION OVERVIEW

   JADI (Nihon boei sobi kogyokai), an industry association (shadan
hojin) of over 130 corporations, promotes the development of a
strong Japanese defense industry.  According to the latest edition
of the annual guide to MITI's public corporations (TSUSHOSANGYOSHO
KANKEI KOEKI HOJIN BENAAN 1993 May 93), JADI seeks to "work towards
the sound development of the defense equipment industry and, in so
doing, contribute to the establishment of Japan's defense base" by
promoting research and development of defense equipment and
improving production technology.  The association's stated
activities regarding defense equipment are as follows:  conducting
surveys and research; gathering and disseminating information,
supporting R&amp;amp;D; offering proposals on modernization, greater
efficiency, and support for Japan's defense production base; hosting
seminars; cooperating with other organizations related to the
defense industry; carrying out contract research, helping devise
Standards; and cooperating with the government's export control
Policies.  Central to the association's activities is the promotion
of R&amp;amp;D and Japan's independent development of defense technology and

product ion capacity (NIKKAN KOGYO SHIMBUN 3 Jun 87, NIKKEI SANGYO
SHIMBUN 10 Jan 87).

   CLOSE TIES TO GOVERNMENT
```

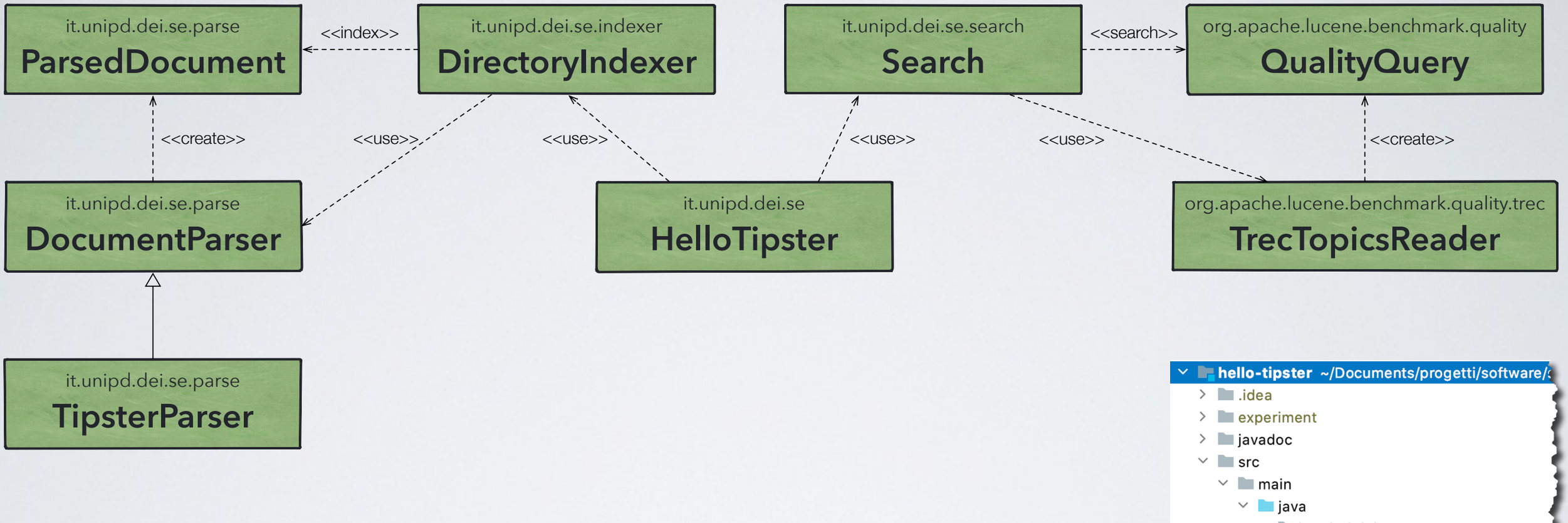
```
<DOC>
<DOCNO> FR940104-0-00001 </DOCNO>
<PARENT> FR940104-0-00001 </PARENT>
<TEXT>

<!-- PJG FTAG 4700 -->

<!-- PJG STAG 4700 -->

<!-- PJG ITAG l=90 g=1 f=1 -->

<!-- PJG /ITAG -->

<!-- PJG ITAG l=90 g=1 f=4 -->
Federal Register
<!-- PJG /ITAG -->

<!-- PJG ITAG l=90 g=1 f=1 -->
&blank;/&blank;Vol. 59, No. 2&blank;/&blank;Tuesday, January 4,
1994&blank;/&blank;Rules and Regulations

<!-- PJG 0012 frnewline -->

<!-- PJG /ITAG -->

<!-- PJG ITAG l=01 g=1 f=1 -->
Vol. 59, No. 2
<!-- PJG 0012 frnewline -->

<!-- PJG /ITAG -->

<!-- PJG ITAG l=02 g=1 f=1 -->
Tuesday, January 4, 1994
<!-- PJG 0012 frnewline -->

<!-- PJG 0012 frnewline -->

<!-- PJG /ITAG -->

<!-- PJG /STAG -->

<!-- PJG /FTAG -->
</TEXT>
</DOC>


<DOC>
<DOCNO> FR940104-0-00002 </DOCNO>
<PARENT> FR940104-0-00001 </PARENT>
<TEXT>

<!-- PJG STAG 4700 -->
```

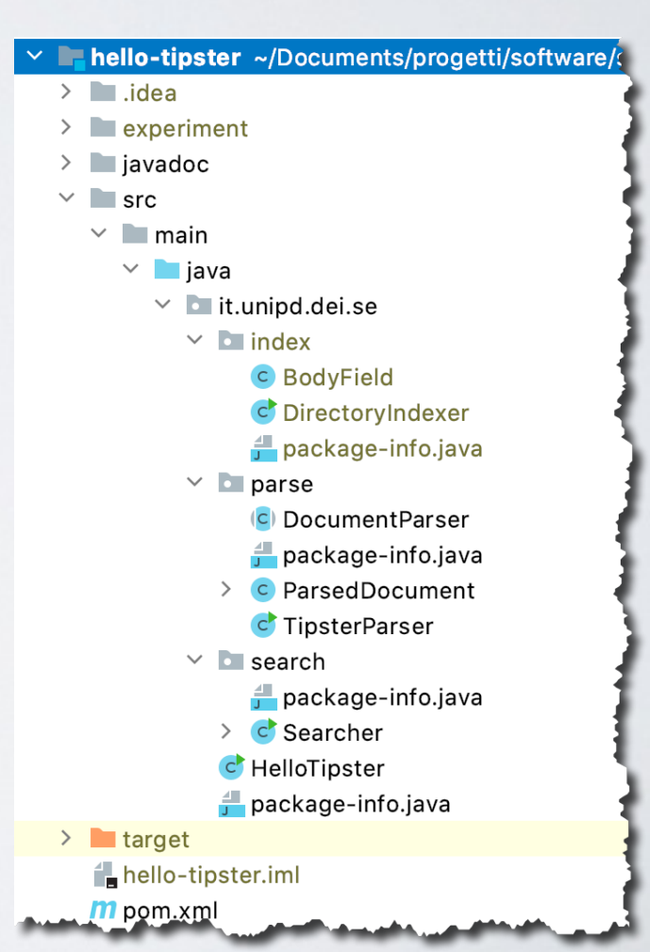
Topics… Do They Really Look Like That?

This is the **topic ID** (only the actual number) you have to use in your `run.txt` file and that you find in `qrels.txt`

```
<top>

<num> Number: 401
<title> foreign minorities, Germany

<desc> Description:
What language and cultural differences impede the integration
of foreign minorities in Germany?

<narr> Narrative:
A relevant document will focus on the causes of the lack of
integration in a significant way; that is, the mere mention of
immigration difficulties is not relevant.  Documents that discuss
immigration problems unrelated to Germany are also not relevant.

</top>

<top>

<num> Number: 402
<title> behavioral genetics

<desc> Description:
What is happening in the field of behavioral genetics,
the study of the relative influence of genetic
and environmental factors on an individual's behavior
or personality?

<narr> Narrative:
Documents describing genetic or environmental factors relating
to understanding and preventing substance abuse and addictions
are relevant.  Documents pertaining to attention deficit disorders
tied in with genetics are also relevant, as are genetic disorders
affecting hearing or muscles.  The genome project is relevant
when tied in with behavior disorders (i.e., mood disorders,
Alzheimer's disease).

</top>

<top>

<num> Number: 403
<title> osteoporosis

<desc> Description:
Find information on the effects of the dietary intakes
of potassium, magnesium and fruits and vegetables as
determinants of bone mineral density in elderly men
and women thus preventing osteoporosis (bone decay).

<narr> Narrative:
A relevant document may include one or more of the
dietary intakes in the prevention of osteoporosis.
```

The Overall Structure





Parsing TIPSTER Documents

```java
public boolean hasNext() {

    String id = null;
    final StringBuilder body = new StringBuilder(BODY_SIZE);

    long lineno = 0;

    try {
        String line;
        Pattern docno_tag = Pattern.compile("<DOCNO>\\s*(\\S+)\\s*<");
        boolean in_doc = false;
        while (true) {
            line = ((BufferedReader) in).readLine();
            lineno++;

            if (line == null) {
                next = false;
                break;
            }
            if (!in_doc) {
                if (line.startsWith("<DOC>")) {
                    in_doc = true;
                } else {
                    continue;
                }
            }
            if (line.startsWith("</DOC>")) {
                in_doc = false;
                body.append(line);
                break;
            }

            Matcher m = docno_tag.matcher(line);
            if (m.find()) {
                id = m.group(1);
            }

            body.append(line).append(" ");
        }

    } catch (IOException e) {
        throw new IllegalStateException("Unable to parse the document.", e);
    }

    if (id != null) {
        document = new it.unipd.dei.se.parse.ParsedDocument(id, body.length() > 0 ?
                body.toString().replaceAll( regex: "<[^>]*>",  replacement: " ") : "#");
    }


    return next;
}
```

- `DocumentParser` is an abstract class which implements the `Iterator` and `Iterable` interfaces
  - `hasNext` tells whether there is another document and `next` returns it, via the `parse` method implemented by the subclasses
- `ParsedDocument` is a plain class for storing the identifier of a document and its body
- TipsterParser does the actual parsing
  - it reads line by line
  - when a new document begins (`<DOC>`), it starts to append lines to the body of the document and extracts the document identifier (`<DOCNO>`)
  - when a document ends (`</DOCNO>`), it replaces all the tags with space
  - further cleaning is left for the subsequent `Analyzer`.

Indexing Documents

```java
*/
public void index() throws IOException {

    System.out.printf("%n#### Start indexing ####%n");

    Files.walkFileTree(docsDir, (SimpleFileVisitor) visitFile(file, attrs) -> {
            if (file.getFileName().toString().endsWith(extension)) {

                DocumentParser dp = DocumentParser.create(dpCls, Files.newBufferedReader(file, cs));

                bytesCount += Files.size(file);

                filesCount += 1;

                Document doc = null;

                for (ParsedDocument pd : dp) {

                    doc = new Document();

                    // add the document identifier
                    doc.add(new StringField(ParsedDocument.FIELDS.ID, pd.getIdentifier(), Field.Store.YES));

                    // add the document body
                    doc.add(new BodyField(pd.getBody()));

                    writer.addDocument(doc);

                    docsCount++;

                    // print progress every 10000 indexed documents
                    if (docsCount % 10000 == 0) {
                        System.out.printf("%d document(s) (%d files, %d Mbytes) indexed in %d seconds.%n",
                                          docsCount, filesCount, bytesCount / MBYTE,
                                          (System.currentTimeMillis() - start) / 1000);
                    }

                }

            }
            return FileVisitResult.CONTINUE;
    });

    writer.commit();

    writer.close();

    if (docsCount != expectedDocs) {
        System.out.printf("Expected to index %d documents; %d indexed instead.%n", expectedDocs, docsCount);
    }

    System.out.printf("%d document(s) (%d files, %d Mbytes) indexed in %d seconds.%n", docsCount, filesCount,
                      bytesCount / MBYTE, (System.currentTimeMillis() - start) / 1000);

    System.out.printf("#### Indexing complete ####%n");
}
```

- `DirectoryIndexer` walks through a tree of directories and subdirectories
- For each file with the requested extension, create a new `DocumentParser` to parse documents within it
- It iterates over the `ParsedDocuments`, transform them into Lucene `Documents`, and index them
- Every 10,000 indexed documents prints progress statistics

Parsing Topics

```java
if (topicsFile == null) {
    throw new NullPointerException("Topics file cannot be null.");
}

if (topicsFile.isEmpty()) {
    throw new IllegalArgumentException("Topics file cannot be empty.");
}

try {
    BufferedReader in = Files.newBufferedReader(Paths.get(topicsFile), StandardCharsets.UTF_8);

    topics = new TrecTopicsReader().readQueries(in);

    in.close();
} catch (IOException e) {
    throw new IllegalArgumentException(
            String.format("Unable to process topic file %s: %s.", topicsFile, e.getMessage()), e);
}

if (expectedTopics <= 0) {
    throw new IllegalArgumentException(
            "The expected number of topics to be searched cannot be less than or equal to zero.");
}

if (topics.length != expectedTopics) {
    System.out.printf("Expected to search for %s topics; %s topics found instead.", expectedTopics,
                      topics.length);
}
```

- The constructor of `Searcher` uses `TrecTopicsReader` to parse the topic file
- The parsed topics are saved in the `topics` array of type `QualityQuery`

Searching Documents

```java
*/
public void search() throws IOException, ParseException {

    System.out.printf("%n#### Start searching ####%n");

    // the start time of the searching
    final long start = System.currentTimeMillis();

    final Set<String> idField = new HashSet<>();
    idField.add(ParsedDocument.FIELDS.ID);

    BooleanQuery.Builder bq = null;
    Query q = null;
    TopDocs docs = null;
    ScoreDoc[] sd = null;
    String docID = null;

    try {
        for (QualityQuery t : topics) {

            System.out.printf("Searching for topic %s.%n", t.getQueryID());

            bq = new BooleanQuery.Builder();

            bq.add(qp.parse(QueryParserBase.escape(t.getValue(TOPIC_FIELDS.TITLE))), BooleanClause.Occur.SHOULD);
            bq.add(qp.parse(QueryParserBase.escape(t.getValue(TOPIC_FIELDS.DESCRIPTION))),
                    BooleanClause.Occur.SHOULD);

            q = bq.build();

            docs = searcher.search(q, maxDocsRetrieved);

            sd = docs.scoreDocs;

            for (int i = 0, n = sd.length; i < n; i++) {
                docID = reader.document(sd[i].doc, idField).get(ParsedDocument.FIELDS.ID);

                run.printf(Locale.ENGLISH, format: "%s\tQ0\t%s\t%d\t%.6f\t%s%n", t.getQueryID(), docID, i, sd[i].score,
                        runID);
            }

            run.flush();

        }
    } finally {
        run.close();

        reader.close();
    }

    elapsedTime = System.currentTimeMillis() - start;

    System.out.printf("%d topic(s) searched in %d seconds.", topics.length, elapsedTime / 1000);

    System.out.printf("#### Searching complete ####%n");
}
```

- For each parsed topic (`QualityQuery`), Searcher creates a Lucene `Query` consisting of the OR of all the terms in the title and description field of the topic
- It then searches the index for that query, retrieving maximum `maxDocsRetrieved` documents
- It process the results list to write the run file

Some Indexing Statistics

	Unique Terms	Size	Time
No stop No stem	1,678,997 the - 523,493 docs of - 509,907 docs in - 501,511 docs and - 500,998 docs to - 497,631 docs	209 MByte	101 seconds
Stop No Stem	1,678,964 from - 370,436 docs page - 346,014 docs has - 320,256 docs which - 302,840 docs have - 294,316 docs	196 MByte	93 seconds
No Stop Stem	1,576,825 the - 523,493 docs of - 509,914 docs in - 501,608 docs and - 501,003 doc to - 497,632 docs	190 Mbyte	109 seconds
Stop Stem	1,576,823 from - 370,436 docs time - 360,749 docs page - 347,260 docs ha - 320,521 docs which - 302,840 docs	177 MByte	112 seconds

Scoring Runs

```
[√ trec_eval-9.0.7 % ./trec_eval ../../collections/TREC_08_1999_AdHoc/qrels.txt ../hello-tipster/experiment/seupd2021-helloTipster-nostop-nostem.txt
runid                   all     seupd2021-helloTipster-nostop-nostem
num_q                   all     50
num_ret                 all     50000
num_rel                 all     4728
num_rel_ret             all     2355
map                     all     0.2151
gm_map                  all     0.1165
Rprec                   all     0.2688
bpref                   all     0.2311
recip_rank              all     0.6983
iprec_at_recall_0.00    all     0.7465
iprec_at_recall_0.10    all     0.4892
iprec_at_recall_0.20    all     0.3485
iprec_at_recall_0.30    all     0.2774
iprec_at_recall_0.40    all     0.2201
iprec_at_recall_0.50    all     0.1840
iprec_at_recall_0.60    all     0.1340
iprec_at_recall_0.70    all     0.0937
iprec_at_recall_0.80    all     0.0668
iprec_at_recall_0.90    all     0.0335
iprec_at_recall_1.00    all     0.0094
P_5                     all     0.4560
P_10                    all     0.4420
P_15                    all     0.4200
P_20                    all     0.3790
P_30                    all     0.3327
P_100                   all     0.1974
P_200                   all     0.1362
P_500                   all     0.0763
P_1000                  all     0.0471
√ trec_eval-9.0.7 %
```

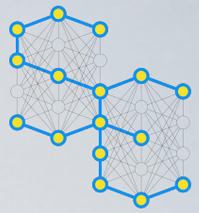
Some Searching Statistics

	Performance	Time
No stop No stem	P@10 - 0.4420 MAP - 0.2151	2 seconds
Stop No Stem	P@10 - 0.44**4**0 MAP - 0.215**2**	2 seconds
No Stop Stem	P@10 - 0.4**880** MAP - 0.2**523**	3 seconds
Stop Stem	P@10 - 0.4**880** MAP - 0.2**532**	2 seconds
Best TREC 8 READWARE2	P@10 - 0.7880 MAP - 0.4692	–
Median TREC 8 UT803b	P@10 - 0.4360 MAP - 0.2598	–
Worst TREC 8 isa25	P@10 - 0.0040 MAP - 0.0026	–

Take Home for the Homework

- Off-the-shelf Lucene provides reasonable performance
  - But to “compete” you will need to go a bit beyond it
- To assess the differences among runs, average performance is not enough. You (will) need to use statistical significance testing
  - Homework 2 will report not only the “raw” performance but also the statistical analyses

What About Parsing Other Document Formats?



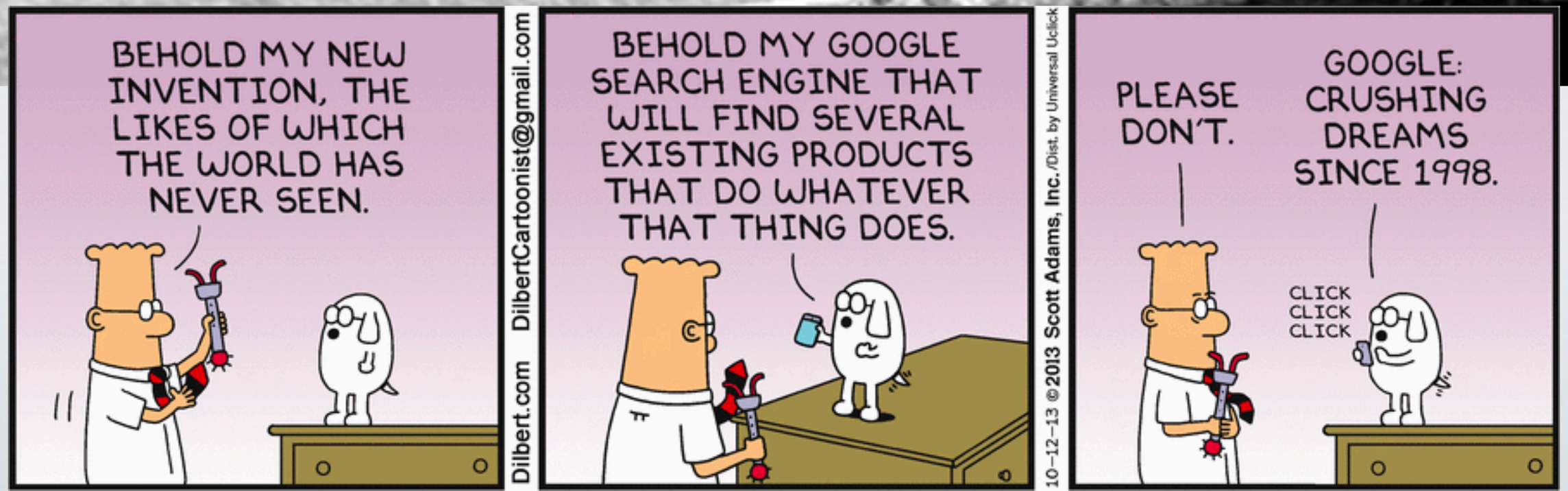
- HTML
  - jsoup — `https://jsoup.org/`
- XML
  - StAX parser part of the Java SE distribution - `javax.xml.stream`
- JSON
  - Jackson - `https://github.com/FasterXML/jackson`
- Popular formats (PPT, XLS, PDF, …)
  - Apache Tika — `https://tika.apache.org/`
- Specific Needs? Write your own parser/lexer
  - JavaCC — `https://javacc.github.io/javacc/`
  - JFlex — `https://jflex.de/`

Exercise

Try it Yourself

- Extend HelloTipster to index the following collections
  - New York Times
  - Washington Post
- Suggestion: you just need to provided an appropriate subclass of **`DocumentParser`**
- Index and search these collections
- Compare your runs to the official TREC ones on these collections

Questions?


BEHOLD MY NEW INVENTION, THE LIKES OF WHICH THE WORLD HAS NEVER SEEN.
Dilbert.com DilbertCartoonist@gmail.com
BEHOLD MY GOOGLE SEARCH ENGINE THAT WILL FIND SEVERAL EXISTING PRODUCTS THAT DO WHATEVER THAT THING DOES.
10-12-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick
PLEASE DON'T.
GOOGLE: CRUSHING DREAMS SINCE 1998.
CLICK CLICK CLICK