On-the-fly Indexing of Large Document Collections

Robert Scholz

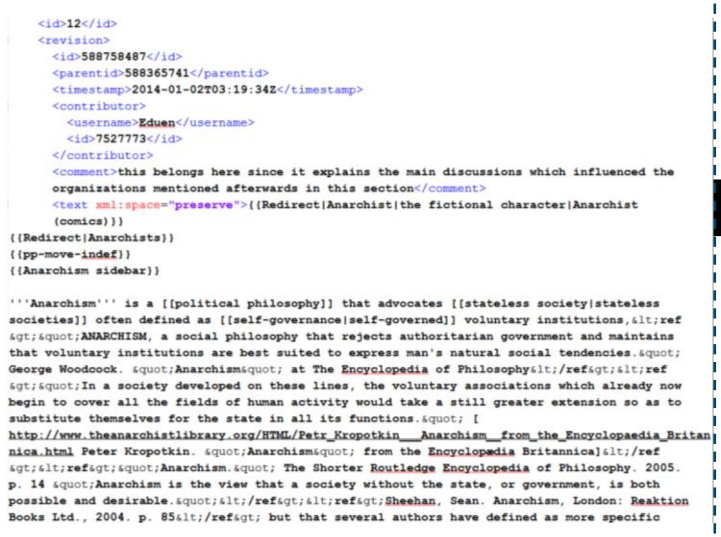
Ekaterina Fuchkina

Outline

- Index review
  - TFIDF
- Dataset structure
- Two different approaches
- Approach 1
- Approach 2
  - UI
- Possible improvements

General concept

Data
XML English Wikipedia
Dump (12 488 908)



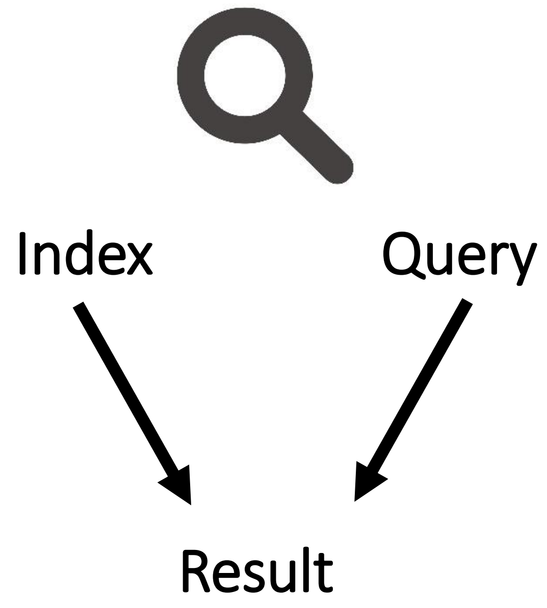
```
    <id>12</id>
    <revision>
      <id>588758487</id>
      <parentid>588365741</parentid>
      <timestamp>2014-01-02T03:19:34Z</timestamp>
      <contributor>
        <username>Eduen</username>
        <id>7527773</id>
      </contributor>
      <comment>this belongs here since it explains the main discussions which influenced the
      organizations mentioned afterwards in this section</comment>
      <text xml:space="preserve">{{Redirect|Anarchist|the fictional character|Anarchist
      (comics)}}
{{Redirect|Anarchists}}
{{pp-move-indef}}
{{Anarchism sidebar}}

'''Anarchism''' is a [[political philosophy]] that advocates [[stateless society|stateless
societies]] often defined as [[self-governance|self-governed]] voluntary institutions,&lt;ref
&gt;&quot;ANARCHISM, a social philosophy that rejects authoritarian government and maintains
that voluntary institutions are best suited to express man's natural social tendencies.&quot;
George Woodcock. &quot;Anarchism&quot; at The Encyclopedia of Philosophy&lt;/ref&gt;&lt;ref
&gt;&quot;In a society developed on these lines, the voluntary associations which already now
begin to cover all the fields of human activity would take a still greater extension so as to
substitute themselves for the state in all its functions.&quot; [
http://www.theanarchistlibrary.org/HTML/Petr_Kropotkin___Anarchism__from_the_Encyclopaedia_Britan
nica.html Peter Kropotkin. &quot;Anarchism&quot; from the Encyclopædia Britannica]&lt;/ref
&gt;&lt;ref&gt;&quot;Anarchism.&quot; The Shorter Routledge Encyclopedia of Philosophy. 2005.
p. 14 &quot;Anarchism is the view that a society without the state, or government, is both
possible and desirable.&quot;&lt;/ref&gt;&lt;ref&gt;Sheehan, Sean. Anarchism, London: Reaktion
Books Ltd., 2004. p. 85&lt;/ref&gt; but that several authors have defined as more specific
```

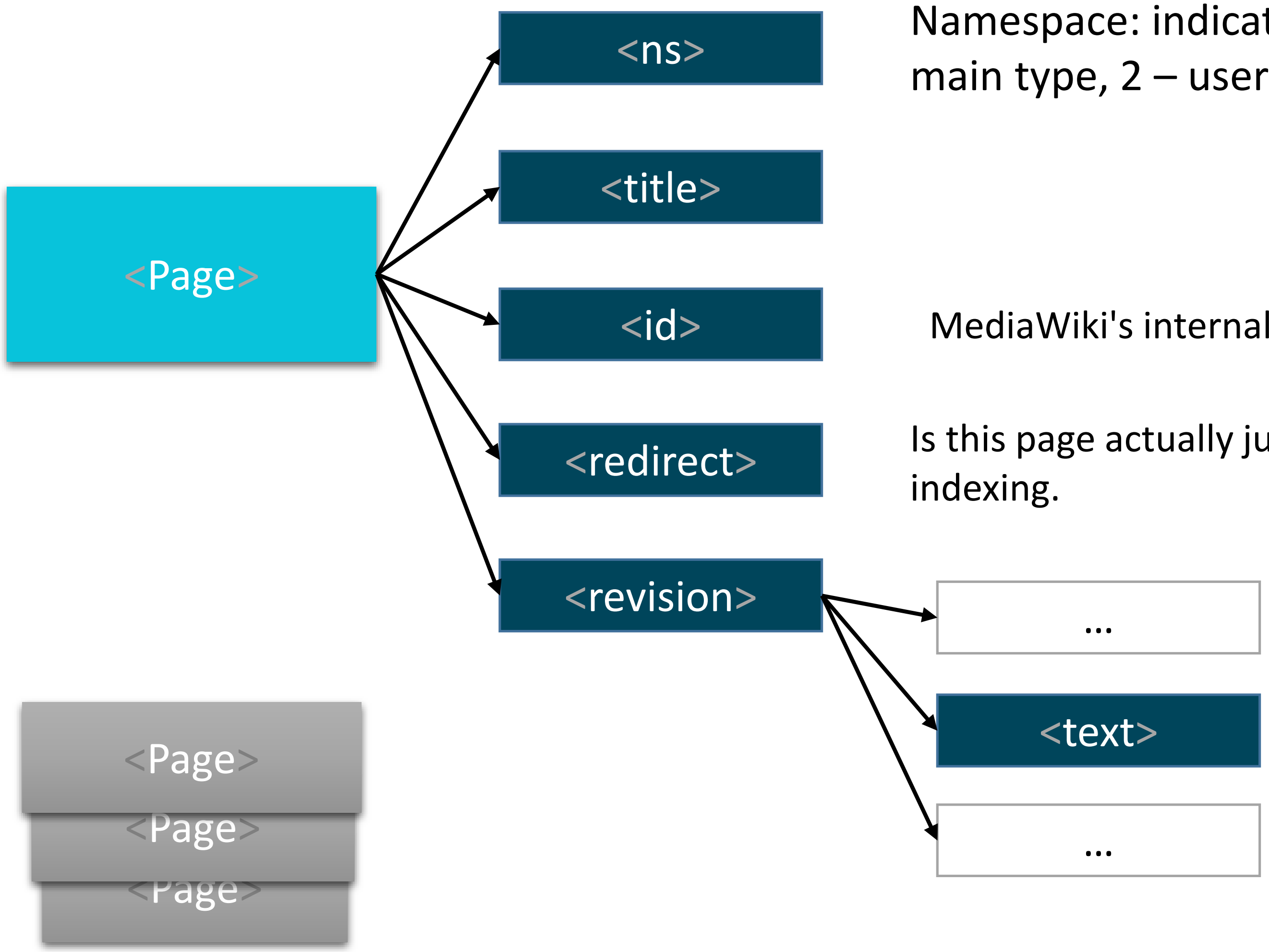
Inverted Index

Term	List of documents ids and TFIDF value			
people	13	1.57	17	1.23
maintain	15	1.83	29	1.56

Search



Dataset structure



Namespace: indicates type of page (e.g. 0 – main type, 2 – user pages. etc.)

MediaWiki's internal article ID

Is this page actually just redirecting to some other? Then exclude it from indexing.

TFIDF

Document Collection
10 million

Some document
100 words

Word “people” appears 3 times
In this document

TF (Term Frequency) of “people” is (3/100) = 0.03

However term “people” appears only in 1000 documents out of 10 million

IDF (Inverse Document Frequency) is log(10000000/1000) = 4

Then, TF*IDF = 0.03 * 4 = 0.12

Two different approaches

- Approach 1: quick and handmade

- Approach 2: using built-in functions

Approach 1

- cd C:\tmp\spark161hd26
- bin\pyspark --packages com.databricks:spark-xml_2.10:0.3.3
- Copy and paste the rest, see comments for intermediate output

```
## count how often each term occurs in a given document
tcnt = pageTokens2.reduceByKey(operator.add)
tcnt.take(2)
# out: [(('when', 25), 2), (('afghanistan', 13), 1)] as in (('word', doc_id),
acc_count_of_word_in_this_doc)


# find the max number of occurences of a single term in a document
max_n_occ_t_per_d = tcnt.map( lambda x : ( x[0][1],x[1])).reduceByKey( lambda a,b : max(a,b) )
max_n_occ_t_per_d.take(5)
#out: [(35, 1), (10, 1), (12, 5), (13, 1), (14, 1)]

##bcmxnocc = sc.broadcast(max_n_occ_t_per_d.collectAsMap())

vvv = tcnt.map(lambda x: (x[0][1],(x[0][0],x[1])))
vvv.take(2)
```

Approach 2

- Usage of pyspark.ml library
  - spark.mllib contains the original API built on top of RDDs.
  - spark.ml provides higher-level API built on top of DataFrames for constructing ML pipelines.
- Methods: HashingTF, IDF, StopWordsRemover
- UI: Django + Spark

Two different approaches

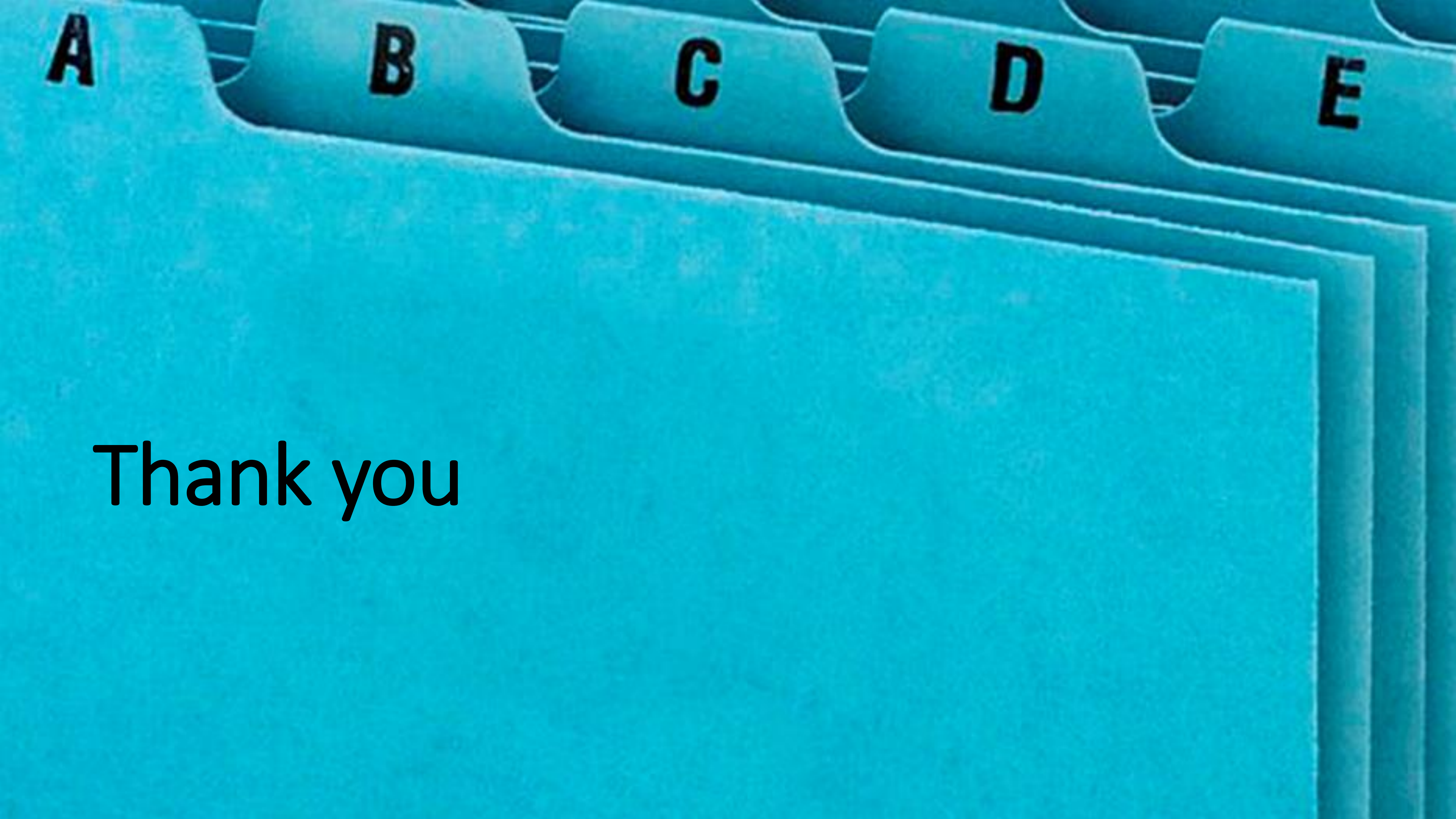
	Approach 1: quick and handmade	Approach 2: using built-in functions
Index build-time	13s / 3h	11 s / 25m
Query time	7 s	11 s
Query	Autism in Afghanistan	Autism in Afghanistan
Output	2 articles	2 articles

Two different approaches

Approach 1: quick and handmade	Approach 2: using built-in functions
+ more flexible (save position of term, context)	+ fast development + compatibility with other library abilities

Possible improvements

- Preprocess XML source (avoid error of malformed nodes)
- N-grams (use other abilities ML library)
- Performance optimization
  - Use in-built hashes (spark scala)
- More sophisticated querying process
- Store the index



Thank you