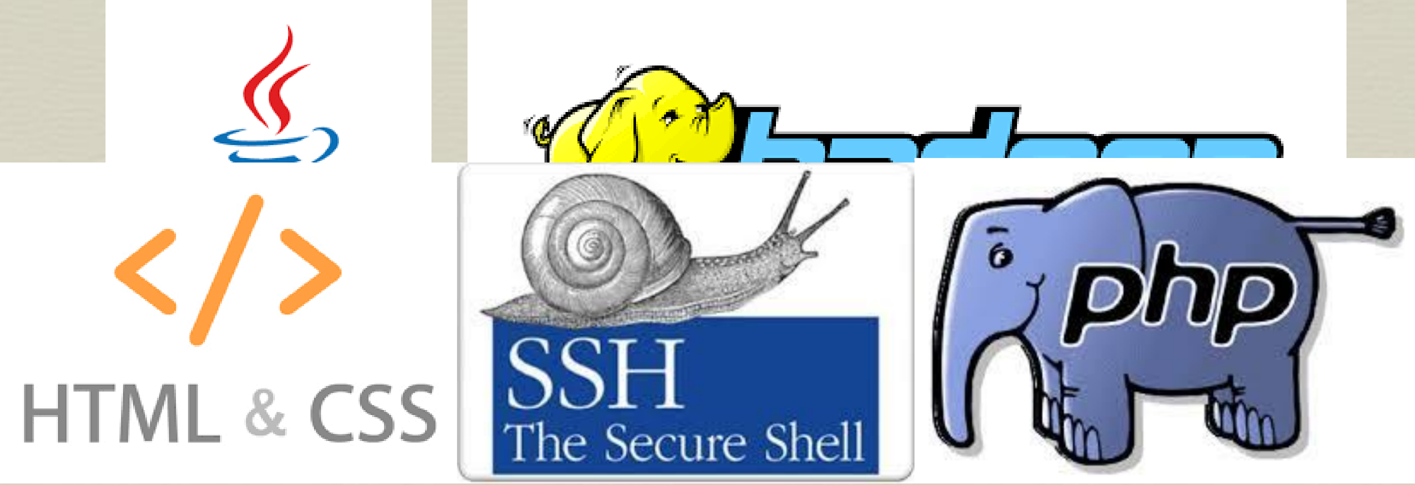
Clemson Search Engine based on MapReduce

Chao Huang

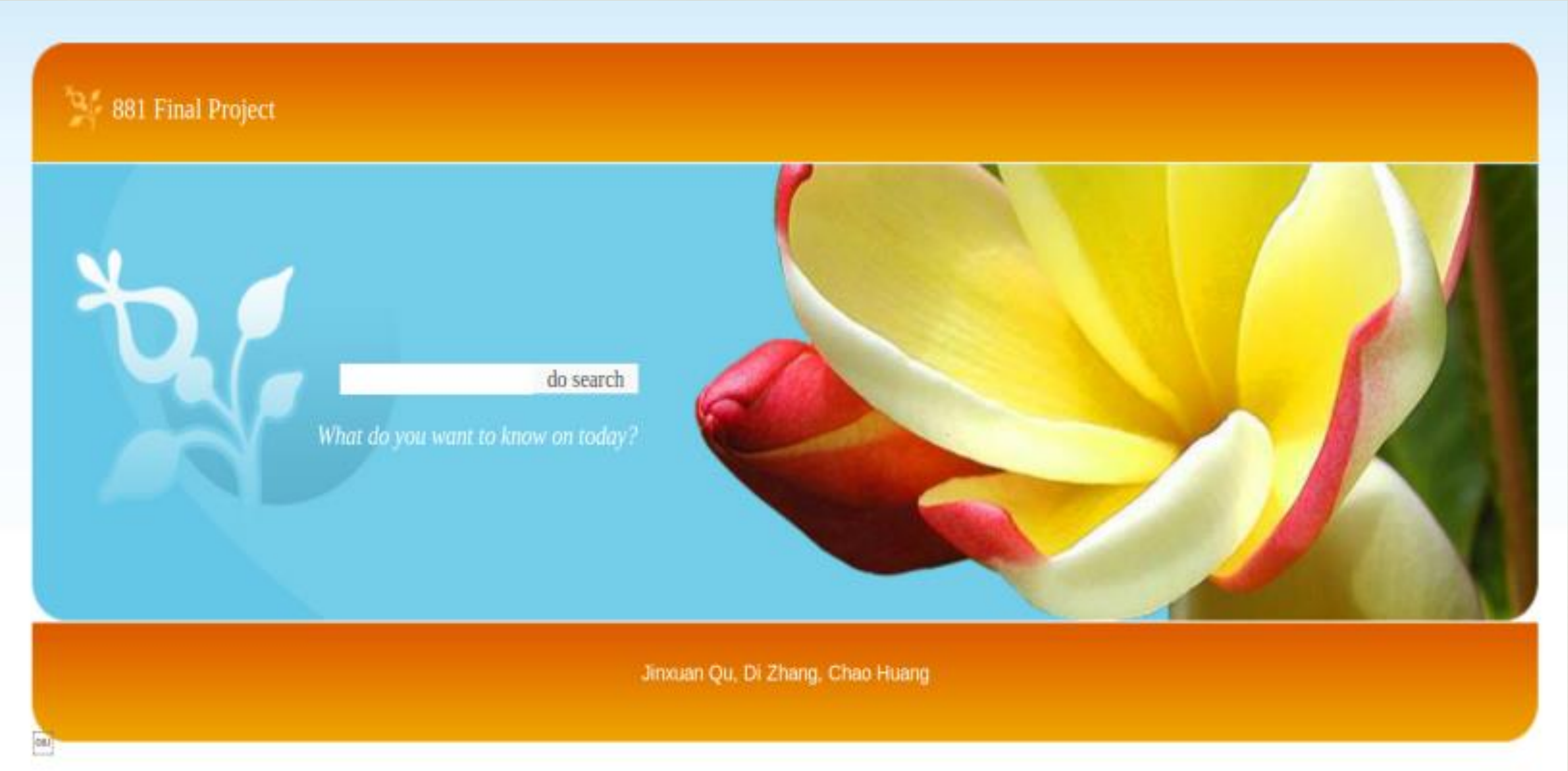
General

- Purpose
  - Implement a Hadoop based IR system
- Data Description
  - Dataset contains 802431 files
  - The Reuter news from 1996.8.20 to 1997.8.29
  - the data format: xml (easy to parse through Java)

Techniques



Search

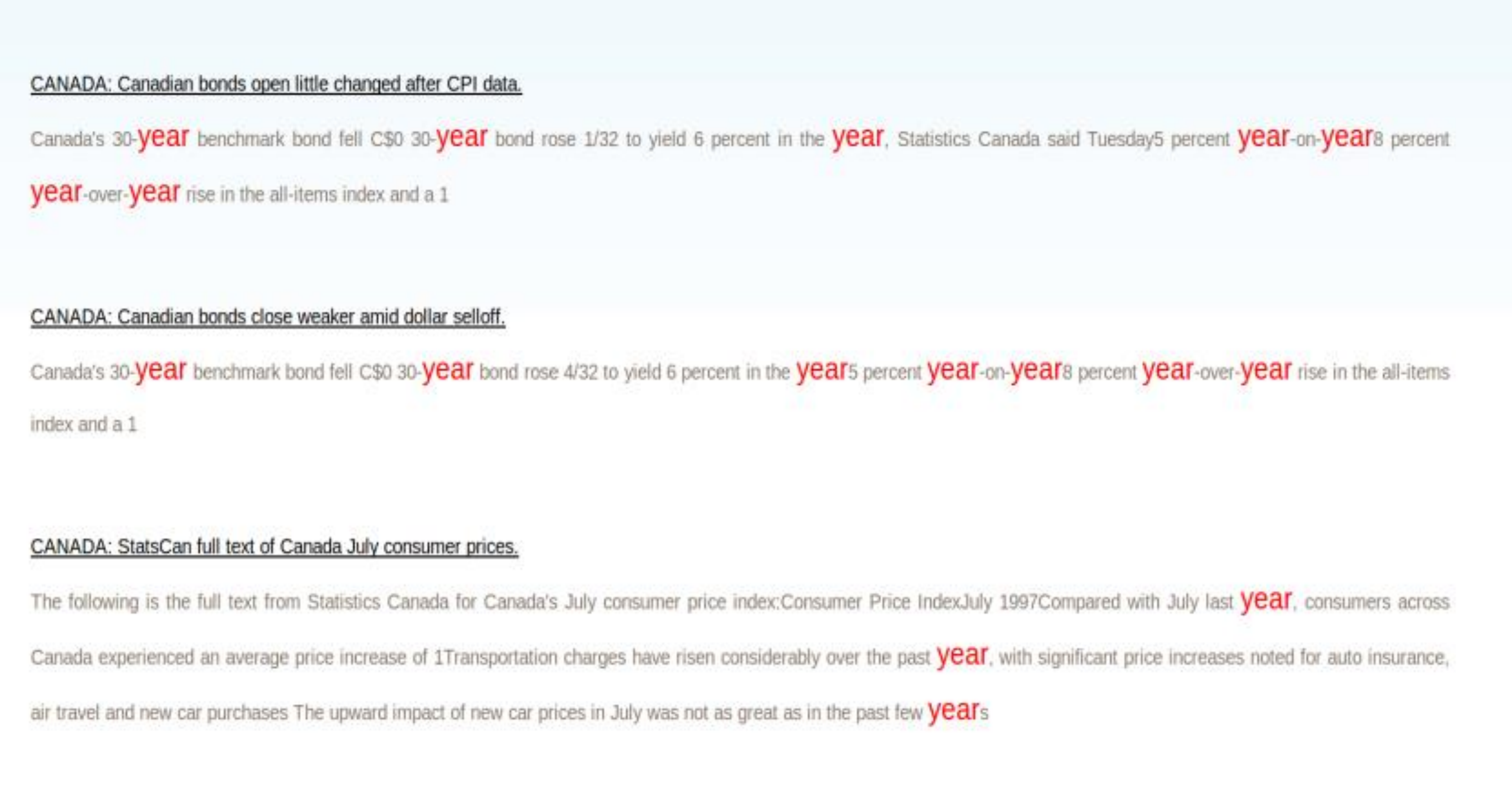


881 Final Project

What do you want to know on today?

Jinxuan Qu, Di Zhang, Chao Huang

Search



CANADA: Canadian bonds open little changed after CPI data.

Canada's 30-year benchmark bond fell C$0 30-year bond rose 1/32 to yield 6 percent in the year, Statistics Canada said Tuesday5 percent year-on-year8 percent year-over-year rise in the all-items index and a 1

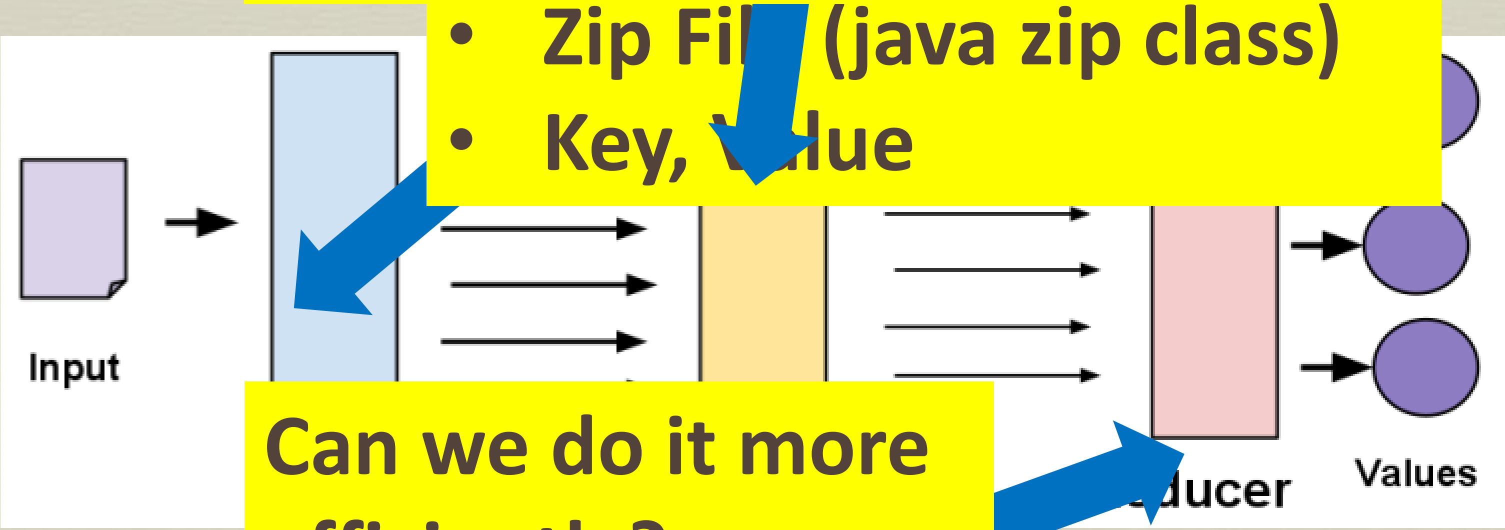
CANADA: Canadian bonds close weaker amid dollar selloff.

Canada's 30-year benchmark bond fell C$0 30-year bond rose 4/32 to yield 6 percent in the year5 percent year-on-year8 percent year-over-year rise in the all-items index and a 1

CANADA: StatsCan full text of Canada July consumer prices.

The following is the full text from Statistics Canada for Canada's July consumer price index:Consumer Price IndexJuly 1997Compared with July last year, consumers across Canada experienced an average price increase of 1Transportation charges have risen considerably over the past year, with significant price increases noted for auto insurance, air travel and new car purchases The upward impact of new car prices in July was not as great as in the past few years

Basic Framework



Basic Hash Partition

- Zip File (java zip class)
- Key, Value

Can we do it more efficiently?

Three factors

- Speed of response
- Size of the index
- Relevance of results

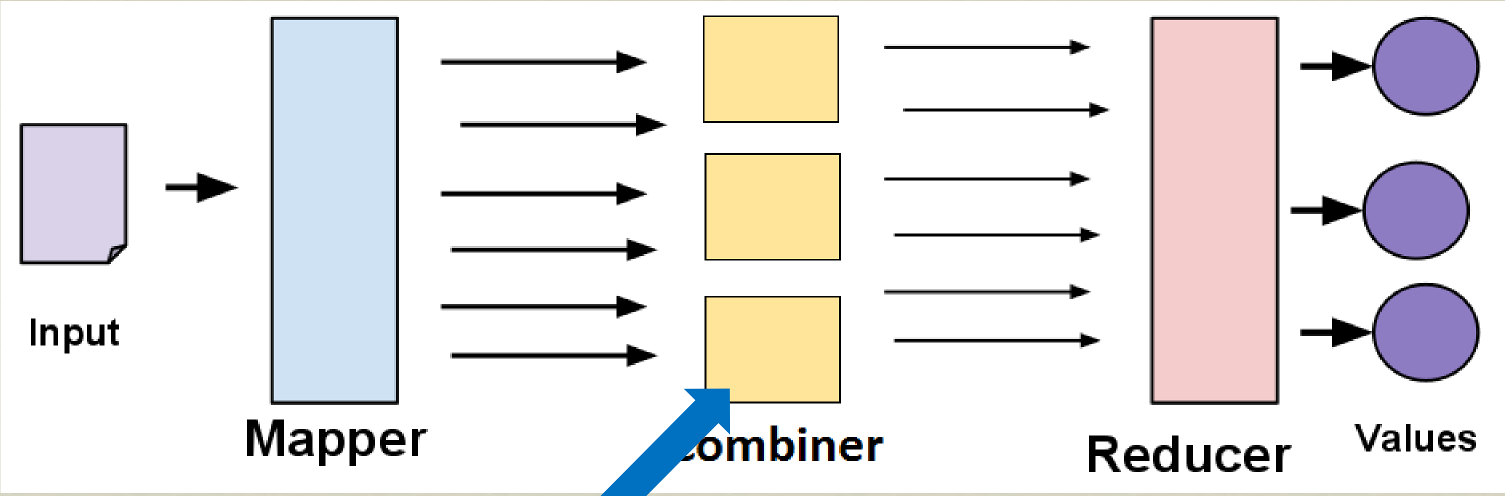
Speed factors

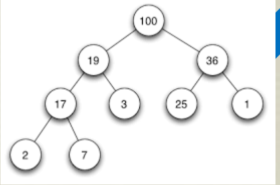
- Speed
- Size
- Relevance

- Different data structure
  - No combiner & hash table
  - Make a heap for each node

Add Combiner

Add a combiner in each node





Different Structure

- Speed
- Size
- Relevance

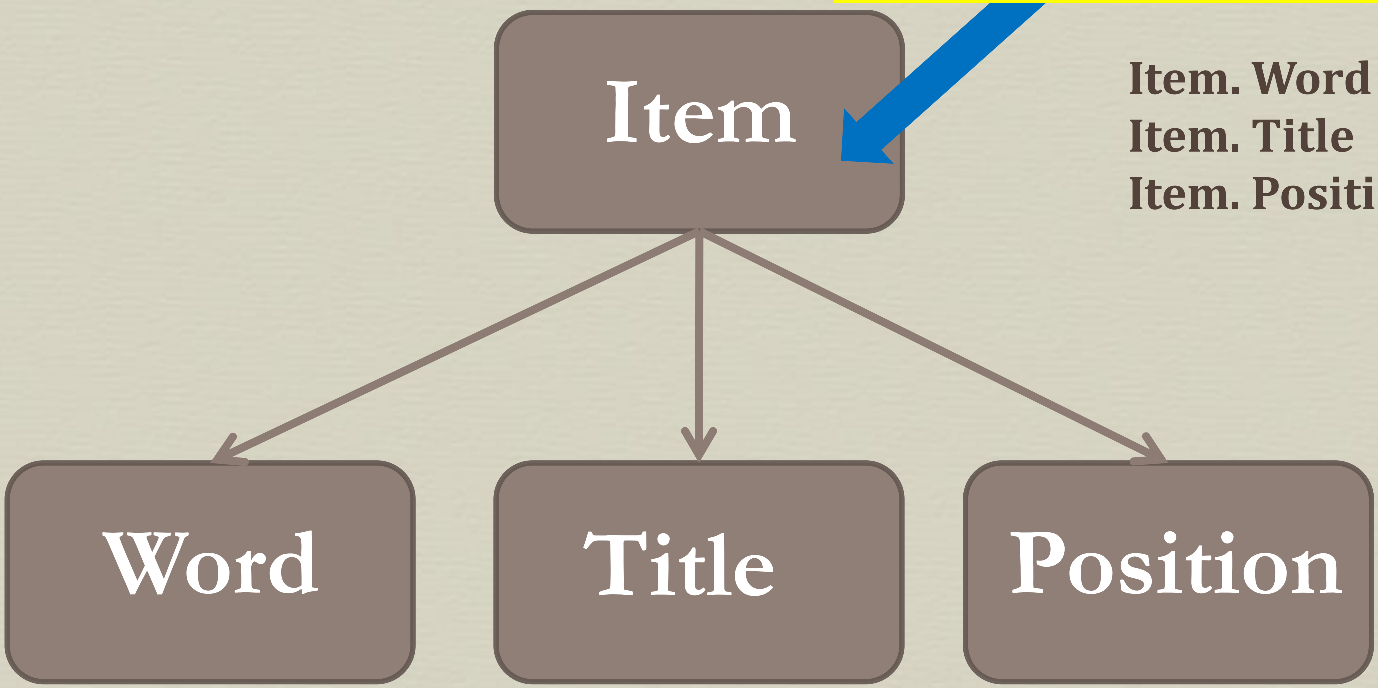
- Different data structure
  - Make a heap for each node
  - Make a new customized class

Problems:

- "word-doc" will be used as the key in mapper
- In the reducer, we have to split this key
- Lots of construction methods will be called

Customize Class

How to make it as a Key



Item. Word
Item. Title
Item. Position

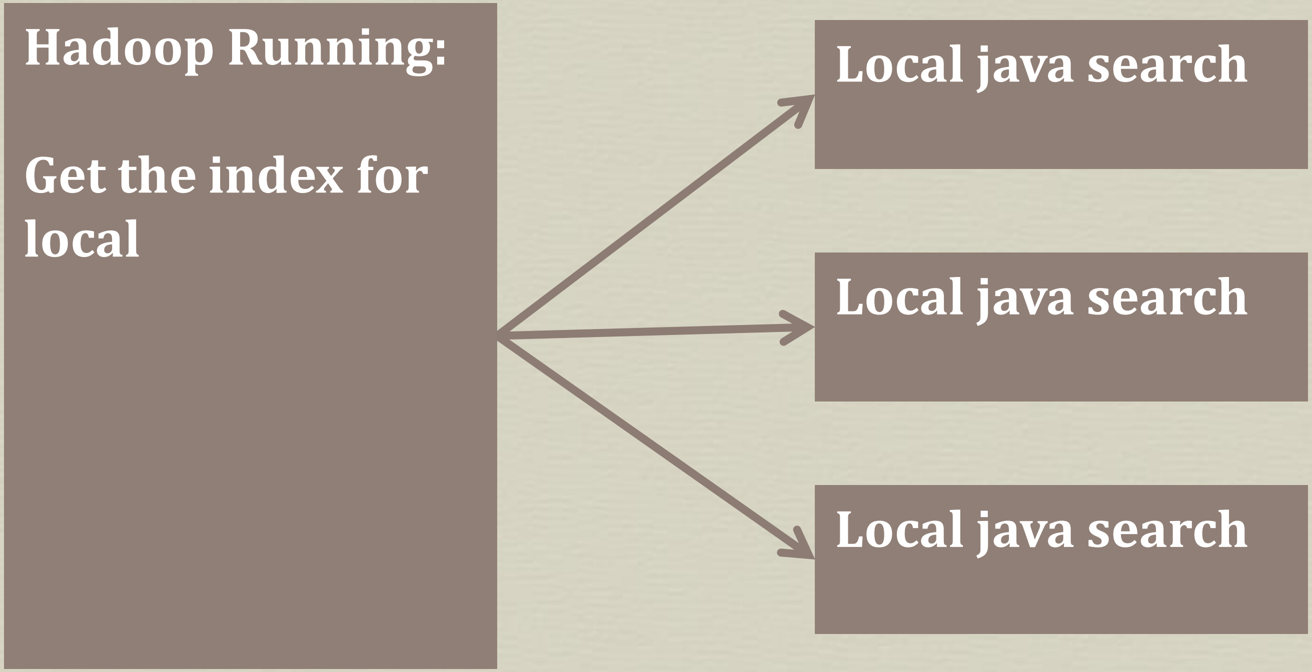
Search Cache

- **Speed**
- Size
- Relevance

- Search Cache (Ajax)
  - Different search strategy
    - 1. search all first
    - 2. step by step

Search All First

- 1. First time search cost longer time



Hadoop Running:

Get the index for local

Local java search

Local java search

Local java search

Search Cache

- Only search for the first page
- Do the search in the backend while reading

Hadoop R

Search fo

Can we do better?

Do next step search

Size of the Index

- Speed of response
- Size of the index
- Relevance of results

Three Index Systems

Word doc1,doc2

Doc1: Count – Doc2:Count

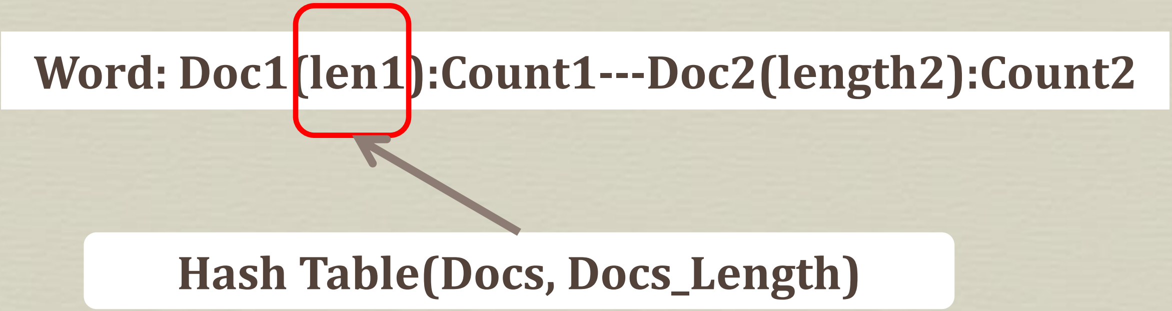
Rank	$W_{t,d} = \left(1 + \log\left(tf_{t,d}\right)\right) * log\,\frac{N}{df_t}$
BM25	$\sum_{t \in q} \log\left[\frac{N}{df_t}\right] * \frac{(k_1+1)tf_{td}}{k_1\left((1-b) + \;\times\left(\frac{L_D}{L_{ave}}\right)\right) + tf_{td}}$

doc1@len:count

Previous Problem

- Speed
- Size
- Relevance

- Single index search
  - Problem:
    - Calculate the length of docs

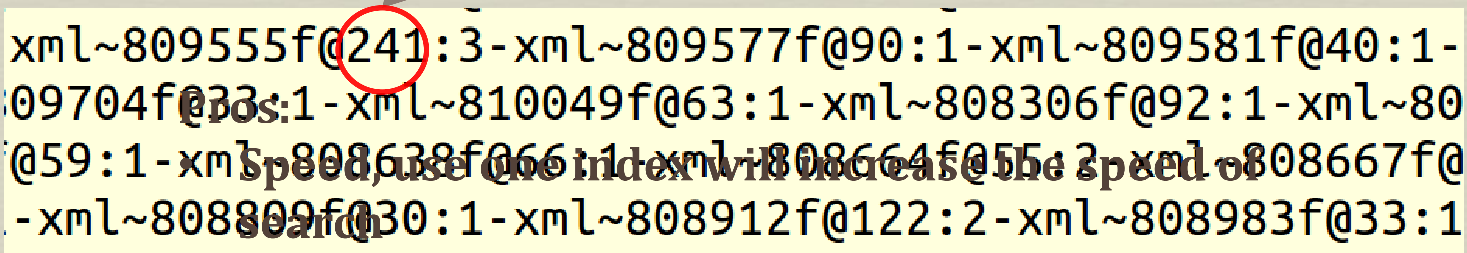


Word: Doc1(len1):Count1---Doc2(length2):Count2

Hash Table(Docs, Docs_Length)

Normal TFIDF Index

Use big hash to get this figure



```
xml~809555f@241:3-xml~809577f@90:1-xml~809581f@40:1-
09704f@33:1-xml~810049f@63:1-xml~808306f@92:1-xml~80
f@59:1-xml~808638f@66:1-xml~808664f@55:2-xml~808667f@
-xml~808809f@30:1-xml~808912f@122:2-xml~808983f@33:1
```

Pros:

- Speed, use the index will increase the speed of search

Cons:

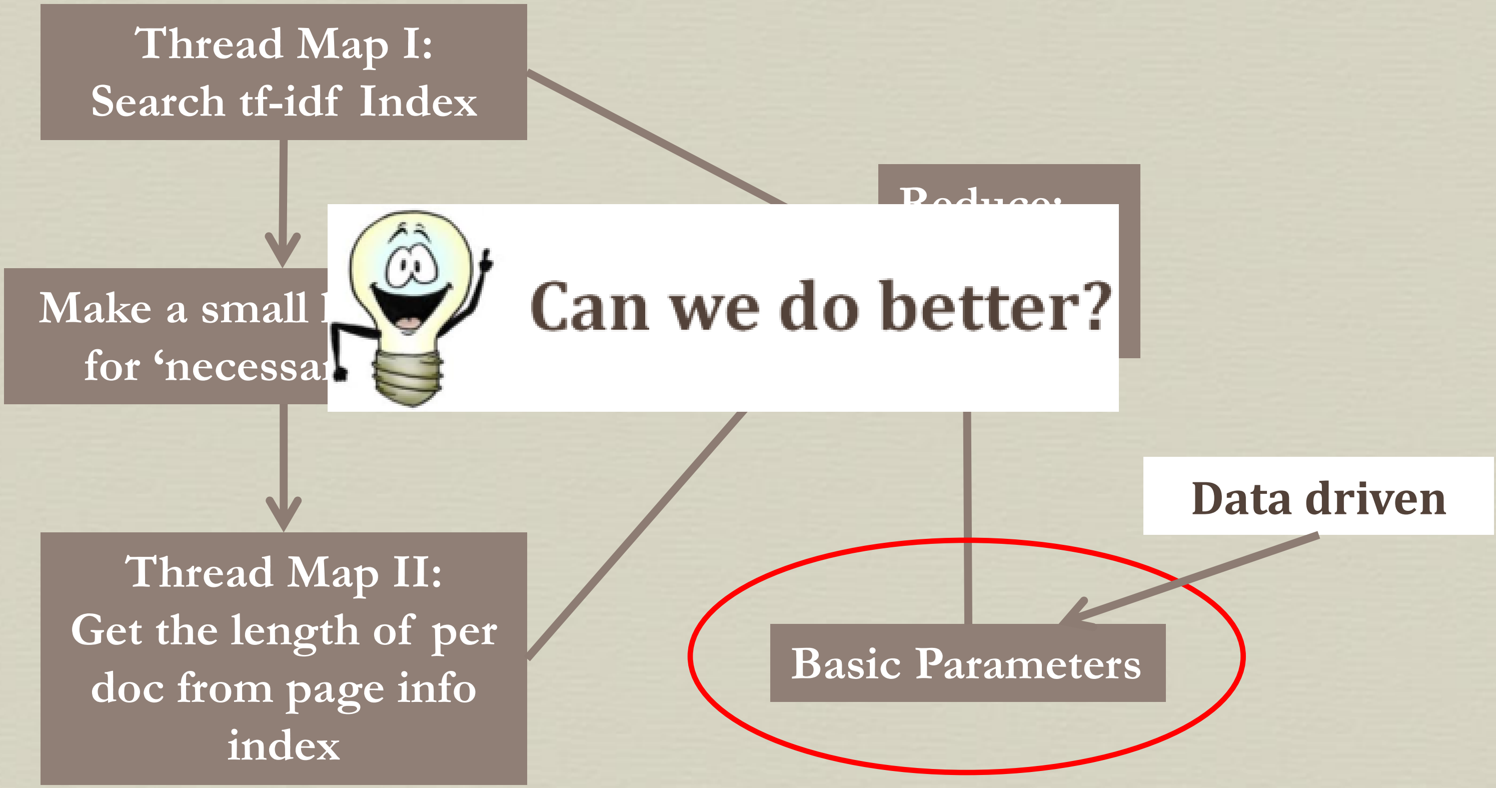
- Big hash table will be used
- Not easy to update the index

Single Index

- Speed
- Size
- Relevance

- Previous: Single index search
  - Problem:
    - Calculate the length of docs
    - Hash table will cost too much space
    - Inconvenient to update Index

Chain Map & Data Driven



Thread Map I:
Search tf-idf Index

Make a small I
for 'necessa

Thread Map II:
Get the length of per doc from page info index

Reduce:

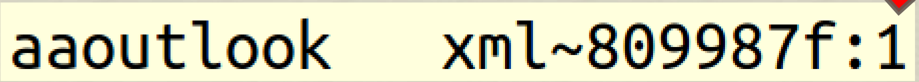
Can we do better?

Data driven

Basic Parameters

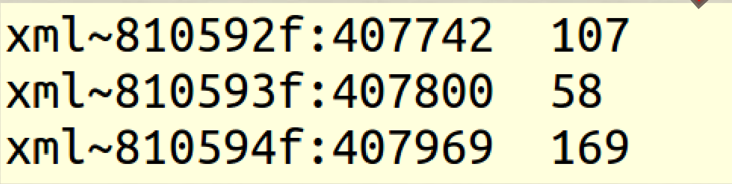
Update Index

- Update index:
  - including insert or delete file



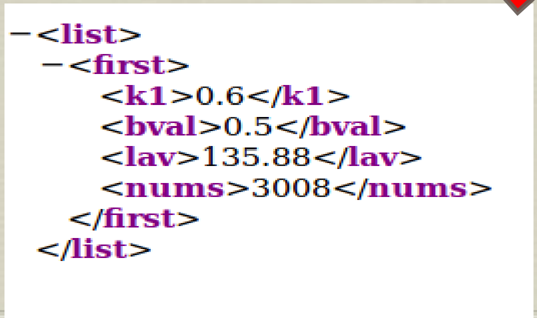
```
aaoutlook    xml~809987f:1
```

Xml~newdoc:2



```
xml~810592f:407742  107
xml~810593f:407800  58
xml~810594f:407969  169
```

Xml~newdoc:408079   100



```
-<list>
  -<first>
      <k1>0.6</k1>
      <bval>0.5</bval>
      <lav>135.88</lav>
      <nums>3008</nums>
    </first>
  </list>
```

Lav = (lav*nums+new)/(nums+1)

Nums ++

Three factors

- Speed of response
- Size of the index
- Relevance of results

Result Measure

- Precision

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(length\ of\ cluster)} = P(\mathrm{re}l\mathrm{e}vant1\mathrm{re}trieved)$$

- Recall

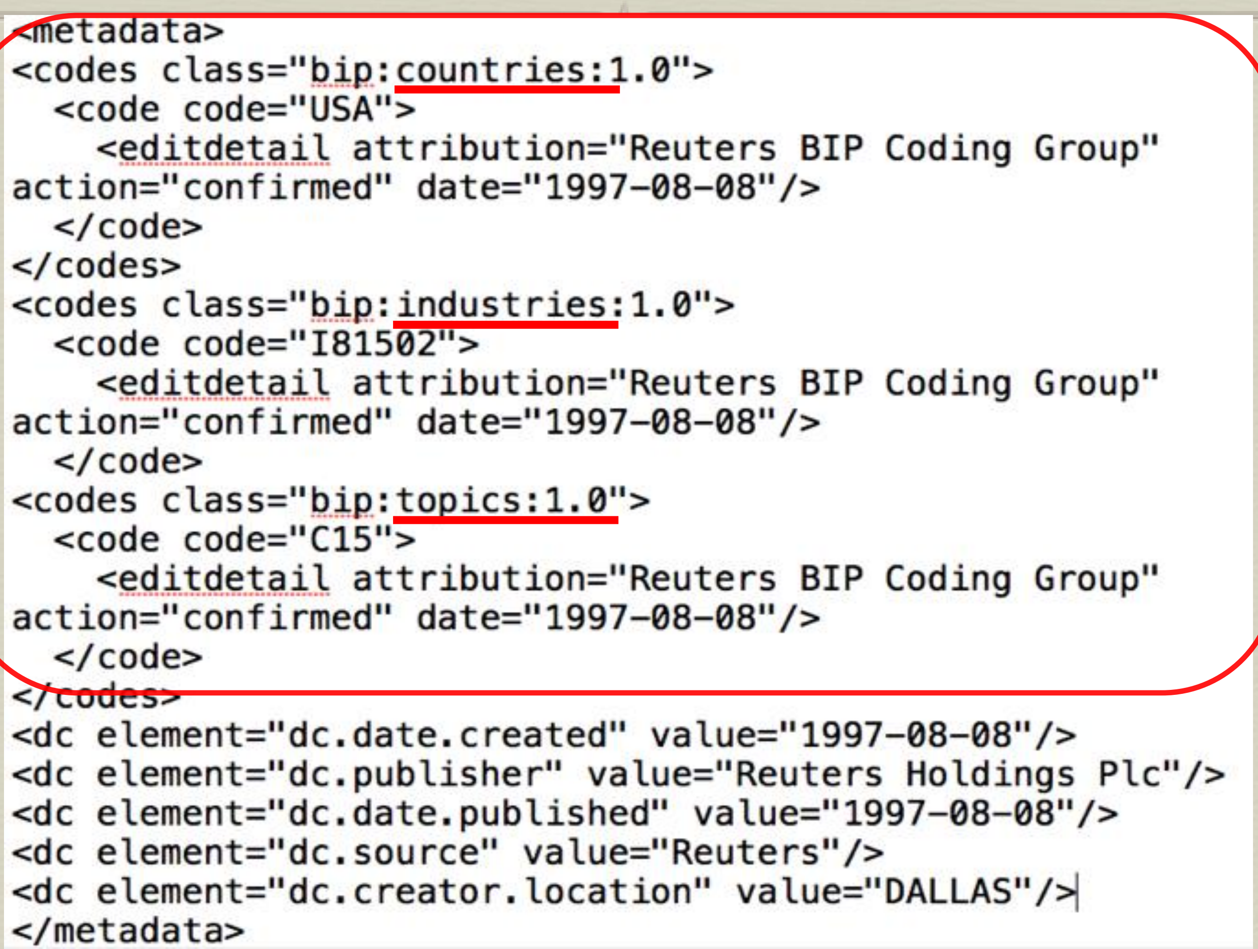
$$\mathrm{Rec}all = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = P(\mathrm{re}trieved1\mathrm{re}l\mathrm{e}vant)$$

User Habit

❖ What a XML file include?

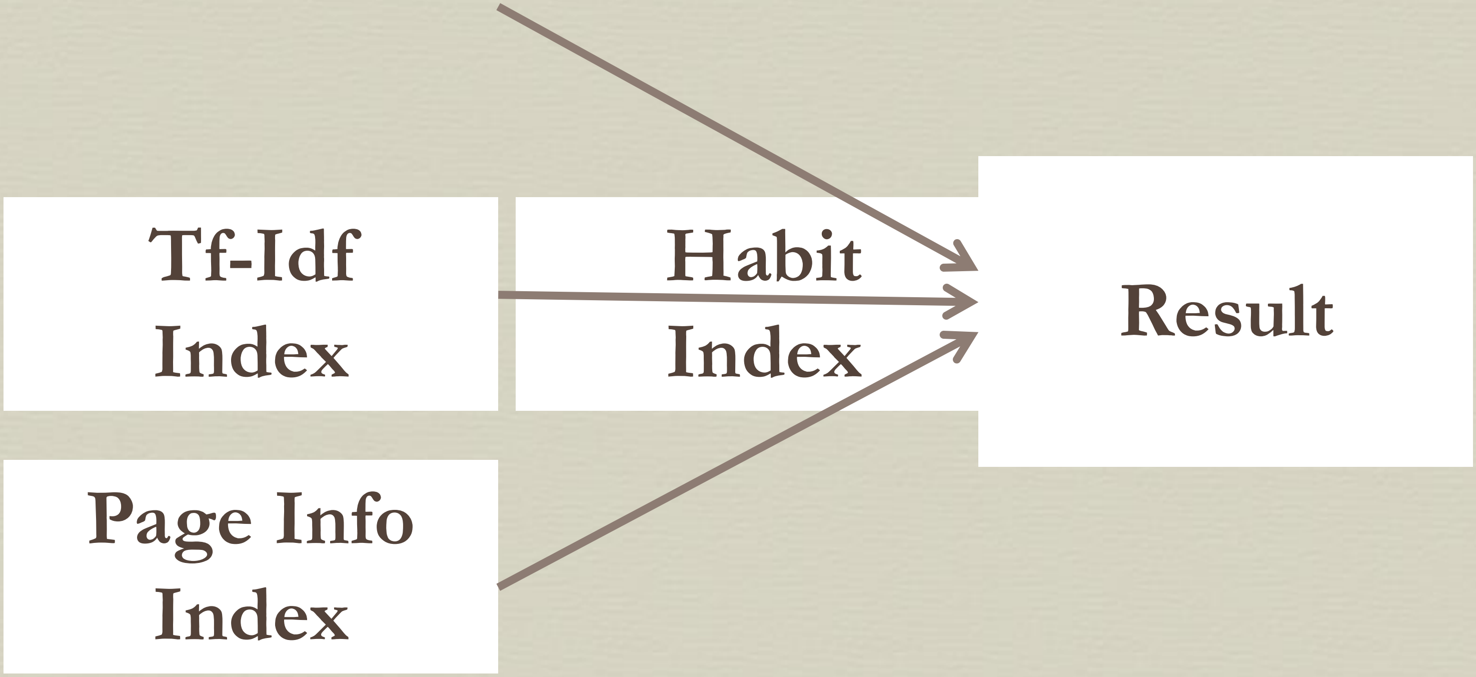
- Title
- Headline
- Dateline
- Text
- Metadata

Something may be ignored



```
<metadata>
<codes class="bip:countries:1.0">
  <code code="USA">
    <editdetail attribution="Reuters BIP Coding Group"
action="confirmed" date="1997-08-08"/>
  </code>
</codes>
<codes class="bip:industries:1.0">
  <code code="I81502">
    <editdetail attribution="Reuters BIP Coding Group"
action="confirmed" date="1997-08-08"/>
  </code>
<codes class="bip:topics:1.0">
  <code code="C15">
    <editdetail attribution="Reuters BIP Coding Group"
action="confirmed" date="1997-08-08"/>
  </code>
</codes>
<dc element="dc.date.created" value="1997-08-08"/>
<dc element="dc.publisher" value="Reuters Holdings Plc"/>
<dc element="dc.date.published" value="1997-08-08"/>
<dc element="dc.source" value="Reuters"/>
<dc element="dc.creator.location" value="DALLAS"/>
</metadata>
```

Create a Region Index



Tf-Idf Index

Habit Index

Page Info Index

Result