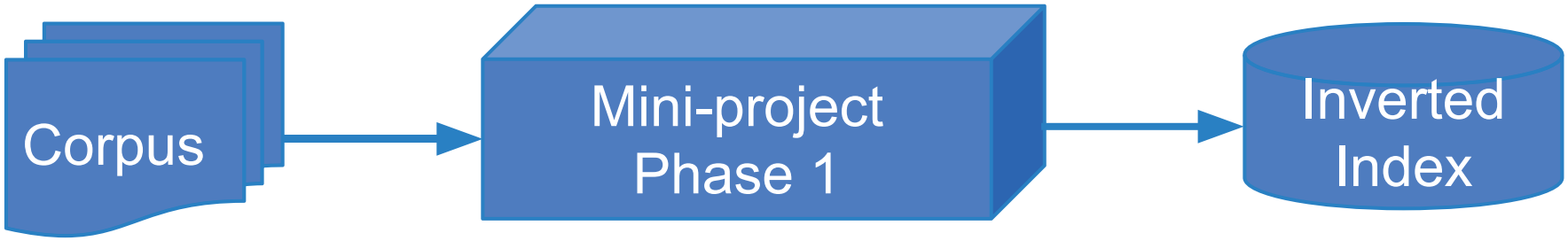
Miniproject

Building a Search Engine

Review

100 MB XML

Parse XML, Tokenization, Case Folding, Stop Word Removal, Stemming, Word Frequency, Posting List / Inverted Index Creation, Optimize



Parsing

SAX Parser (**S**imple **A**PI for **X**ML)

Why not DOM?

Using WikiXMLj is not allowed..

Tokenization

“The time of the Elves… is over. Do we leave Middle-Earth to its fate? Do we let them stand alone?” ~ LOTR

Middle-Earth v/s **Middle** and **Earth**

1984 (George Orwell)
O’Neill - neill, oneill, o’neill, o’ neill, o neill

Rules are Language Specific
Lebensversicherungsgesellschaftsangestellter

Tokenization

Split String

- Blanks and Special Characters may act as delimiters

Numbers may or may not be considered

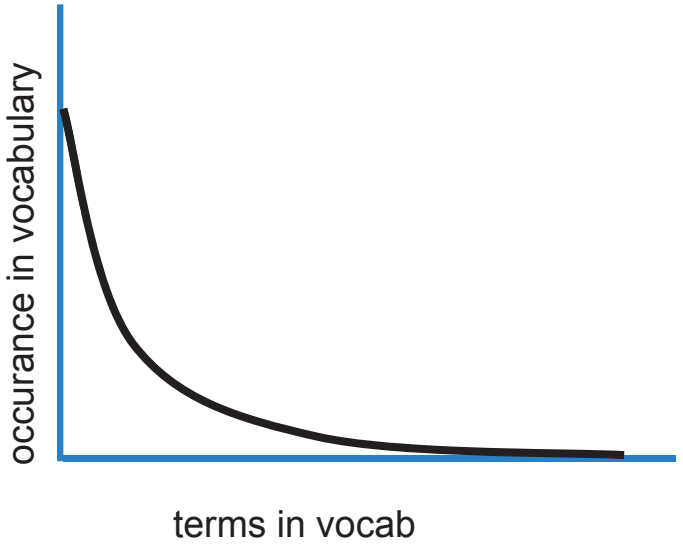
- Depends on the purpose

Dropping Stop Words

Why to Drop?

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

There are many more such words



Dropping Stop Words

Fail

Let It Be

To Be Or Not To Be

```
The time of the
Elves… is over. Do
we leave Middle-Earth
to its fate? Do we
let them stand alone?
```

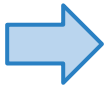
```
time Elves over leave
Middle Earth fate
stand alone
```

Normalization

Stemming and Lemmatization

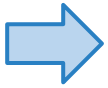
Create Equivalence Classes of Terms

```
operate operating operates
operation operative
operatives operational
```



oper

```
is, am, are
```



be

Normalization

Stemming and Lemmatization

Porter Stemming Algorithm

```
sses -> ss      caresses -> caress
ies -> i        ponies -> pony
ss ->ss         caress -> caress
s -> _          cats -> cat
```

To learn more, visit http://tartarus.org/martin/PorterStemmer/

or simply google “porter stemming algorithm”

Inverted Index

DOC 1

But I am the real Strider, fortunately. I am Aragorn son of Arathorn; and if by life or death I can save you, I will. I am real.

real strider fortunately aragorn son arathorn life death save real

real strider fortun aragorn son arathorn life death save real

real	2
strider	1
fortun	1
aragorn	1
son	1
aranthorn	1
life	1
death	1
save	1

Inverted Index

DOC 2

Many that live deserve death. And some that die deserve life. Do not be too eager to deal out death in judgement.

live deserve death die deserve life eager deal death judgement

live deserv death die deserv life eager deal death judgement

live	2
deserv	2
death	2
die	1
life	1
eager	1
deal	1
judgement	1

Inverted Index

real	2
strider	1
fortun	1
aragorn	1
son	1
aranthorn	1
life	1
death	1
save	1

DOC 1

live	2
deserv	2
death	2
die	1
life	1
eager	1
deal	1
judgement	1

DOC 2

aragon : d1 (1)
aranthorn : d1 (1)
deal : d2(1)
death : d1(1), d2(2)
deserv : d2(2)
die : d2 (1)
eager : d2(1)
fortun : d1 (1)
judgement : d2(1)
life : d1 (1), d2(1)
live : d2 (2)
real : d1 (2)
save : d1 (1)
son: d1(1)
strider : d1 (1)

Handling Multiple Fields

Wikipedia Fields:-

1. Title
2. Body Text
3. Infobox
4. Categories
5. External Links (outlinks)
6. References

Handling Multiple Fields

```
{{Infobox Tolkien character
| image_character = Sauron Tolkien illustration.jpg
| image_caption   = [[J. R. R. Tolkien]]'s watercolour illustration of Sauron.
| character_name  = Sauron
| character_alias = See [[#Names and titles|Names and titles]] below
| character_race  = [[Ainur (Middle-earth)|Ainur]]
| Book(s)         = {{Plainlist |
* "[[The Hobbit]]"
* "[[The Lord of the Rings]]"
* "[[The Silmarillion]]"
* "[[Unfinished Tales]]"
* "[[The Children of Húrin]]"}}
}}
```

Handling Multiple Fields

Multifield 1:

```
sachin:d1-t1c2b7|d5-t1
tendulkar:d1-t1b1|d6-c1b1
```

Multifield 2:

```
sachin-t:d1-1|d5-1
sachin-c:d1-2
sachin-b:d1-7
tendulkar-t:d1-1
tendulkar-c:d6-1
tendulkar-b:d1-1|d6-1
```

Handling Multiple Fields

Multifield 1:

```
sachin:d1-t1c2b7|d5-t1
tendulkar:d1-t1b1|d6-c1b1
```

Multifield 2:

```
sachin-t:d1-1|d5-1
sachin-c:d1-2
sachin-b:d1-7
tendulkar-t:d1-1
tendulkar-c:d6-1
tendulkar-b:d1-1|d6-1
```

For this task, use Wikipedia Page ID as document ID

Handling Multiple Fields

Example: `t:Sachin b:Tendulkar c: Sports i:Mumbai`

`t:Sachin l:IPL r:times`

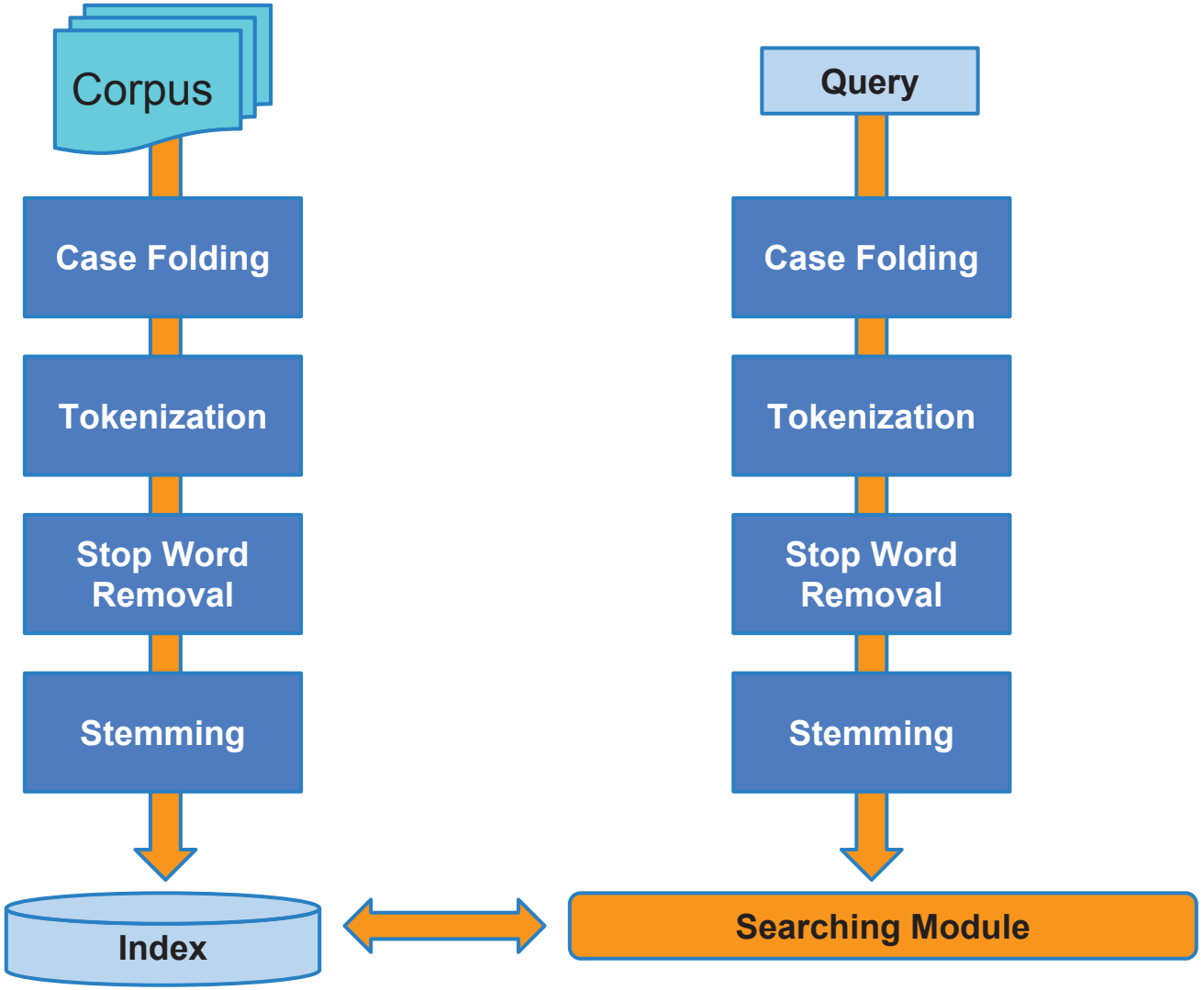
`Sachin`

Store type along with frequency and pageid

Search can be either field based or plain - not explicit.

`Example: Sachin Tendulkar Sports`

Searching


Corpus
Case Folding
Tokenization
Stop Word Removal
Stemming
Index
Query
Case Folding
Tokenization
Stop Word Removal
Stemming
Searching Module

Evaluation

- Automated Evaluation
  - Follow the Guidelines

Submission Format:

Create a compressed file including your project files in the following structure:-

```
201101xxx_as1.tar.gz
201101xxx_as1/
bin/index.sh
bin/query.sh
src/*
readme.txt
```

Evaluation

index.sh

This script invokes your Indexing engine, takes two parameters, one for the corpus XML file, and other for the location of the index folder.

```
Usage: bash index.sh ~/data/sample.xml ~/index_folder
```

query.sh

This script takes from stdin, the number of queries on the first line, and a separate search term in each line, and prints the list of comma separated document-ids(as extracted from the dump) in sorted order per line, for that query.

```
Usage: printf “2\nSachin Tendulkar\ngandalf\n” | bash query.sh
```

Stay tuned for further assignments.