



Searching & Indexing Text Data & Beyond

Michael Stewart

The Plan

- Web search engines

- Aspects of text
- Ranking
- Setting up Lucene

- Non-text data

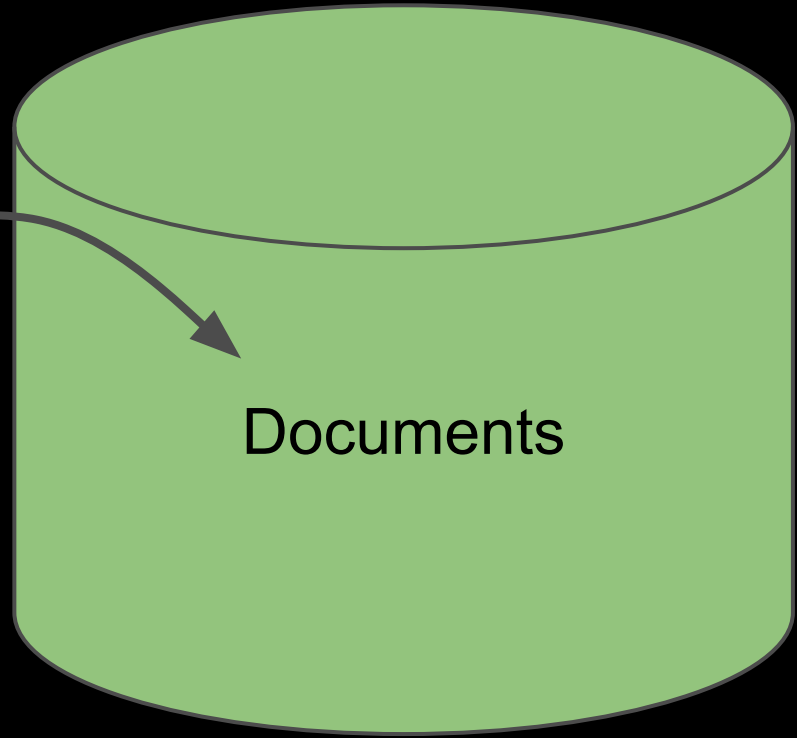
time permitting:

- Alternate search paradigms
 - User-driven exploration

Not covered

- sharding
- hadoop
- advanced text processing

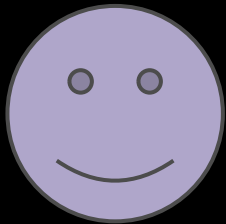
When would I use a search engine?



Not well structured: ~~SQL~~

Web search

Design issue: What does the user want?



Retrieval?

Exploration?

Web search

People issues:

Adversaries

Exploitation?

Evil Spiders

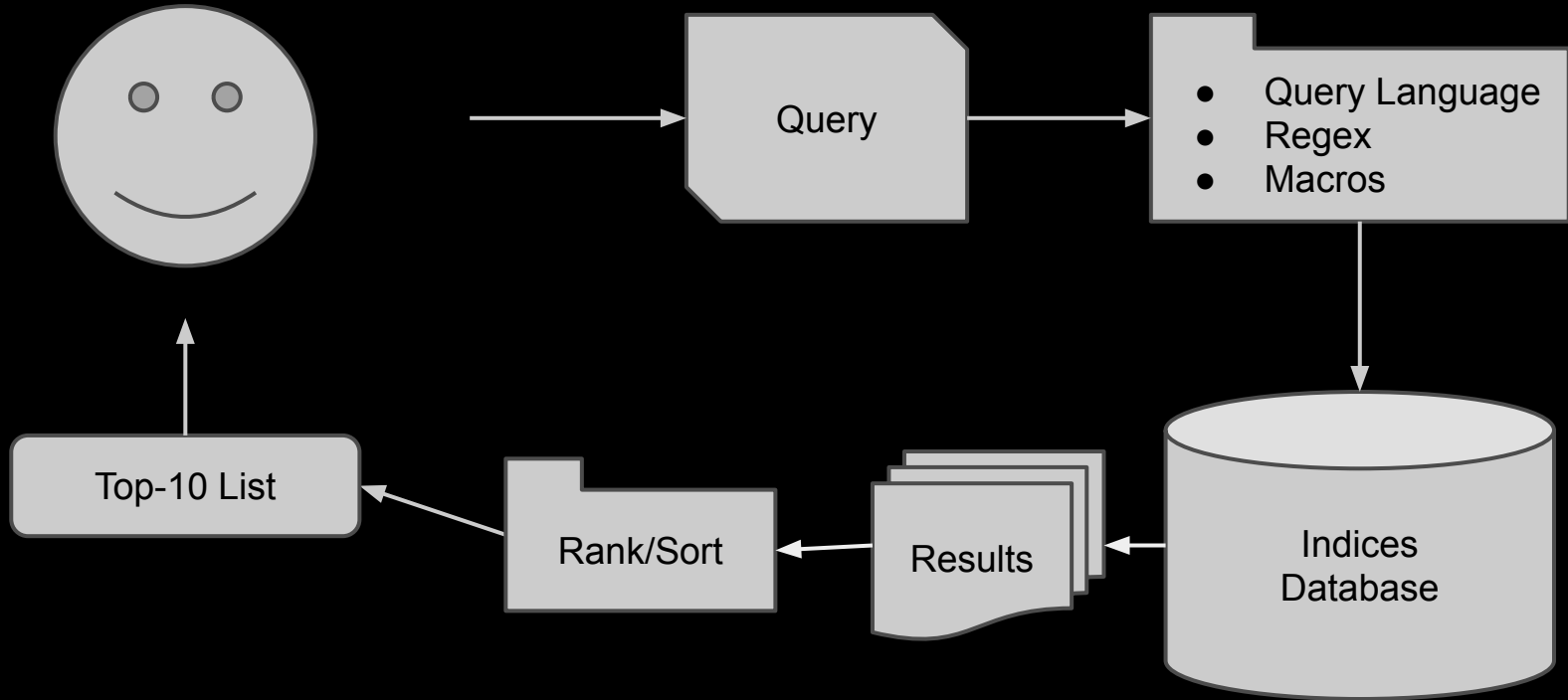
Remember when you threw a shoe at me and I fell somewhere behind the bed?



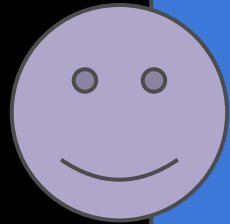
I remember too.



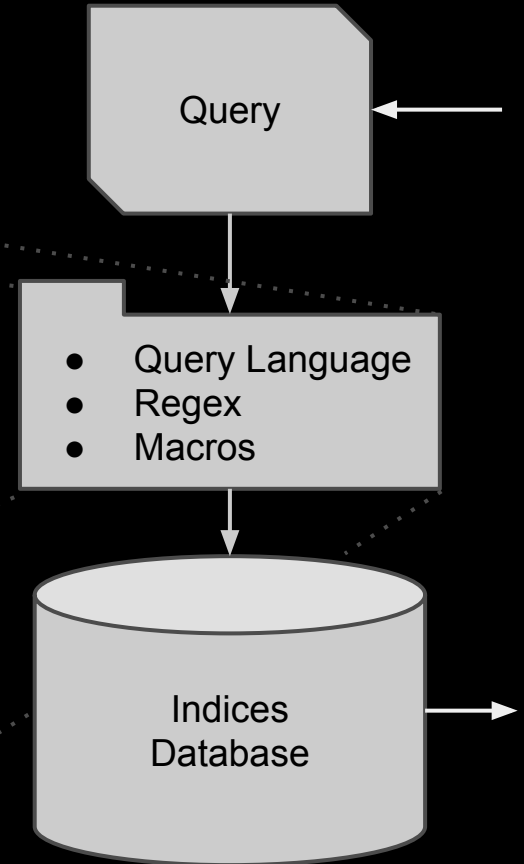
Web search



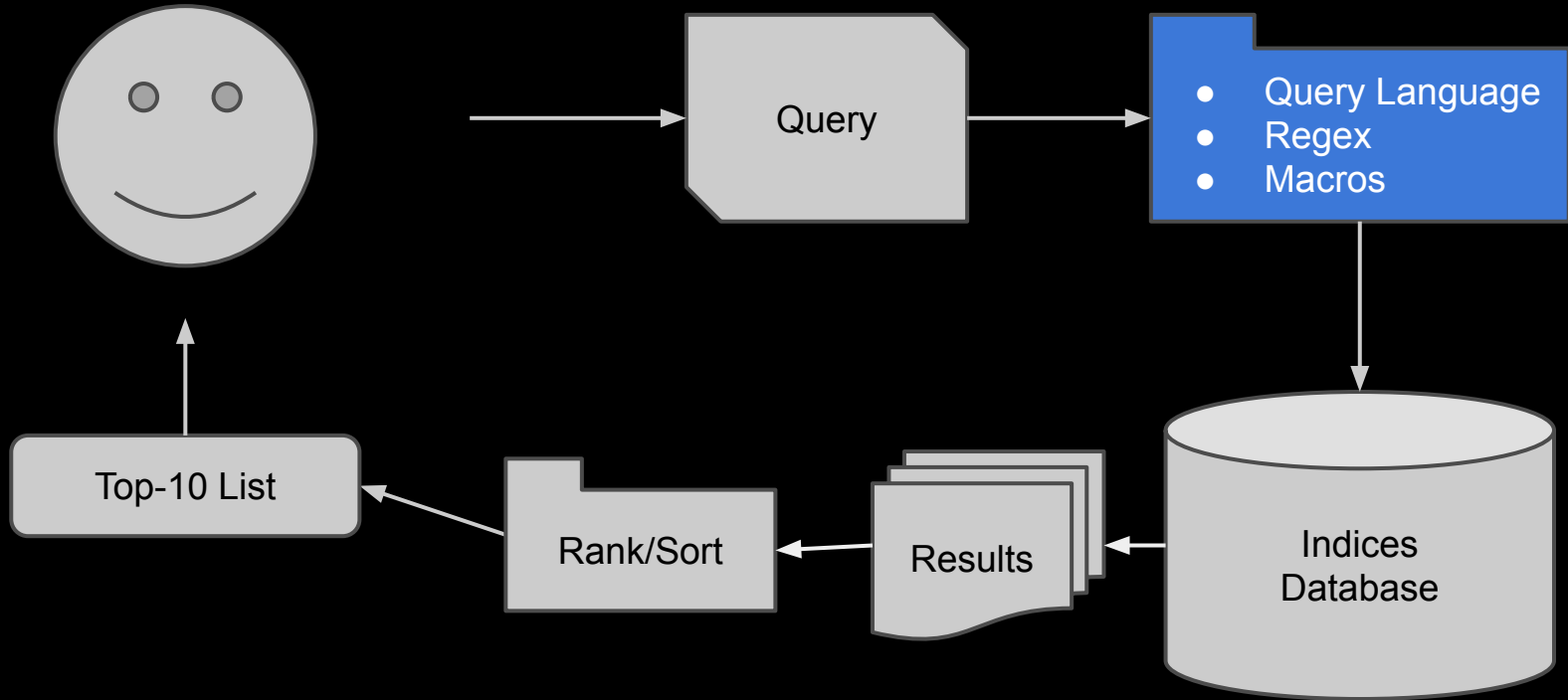
Web search



Fuzzy search
Personalization
Instant results, feedback



Web search



Web search - What's an Index?

Usually a vector-space mapping.

Doc+Term \rightarrow Score

e.g. Bag of words frequency

	Doc 1	Doc 2	Doc 3
Word 1	5	8	8
Word 2	2	0	6
Word 3	1	7	0

On disk
sparse matrix:

```
1 1 5
1 2 8
1 3 8
2 1 2
2 3 6
3 1 1
3 2 7
```

Web search - What's an Index?

"Block Sort-Based Indexing"

Others support features e.g. [RegEx](#)

	Doc 1	Doc 2	Doc 3
Word 1	5	8	8
Word 2	2	0	6
Word 3	1	7	0

Indices

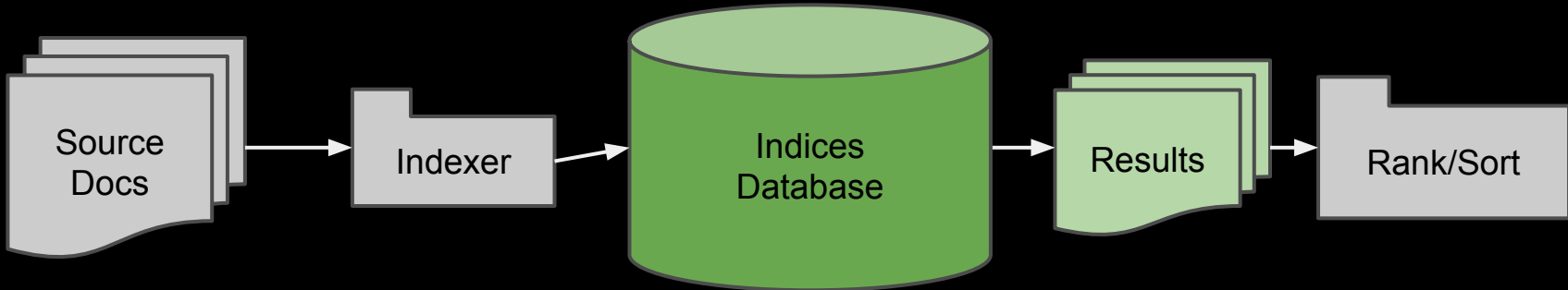


	TF	DF
Word 1	21	3
Word 2	8	2
Word 3	8	2

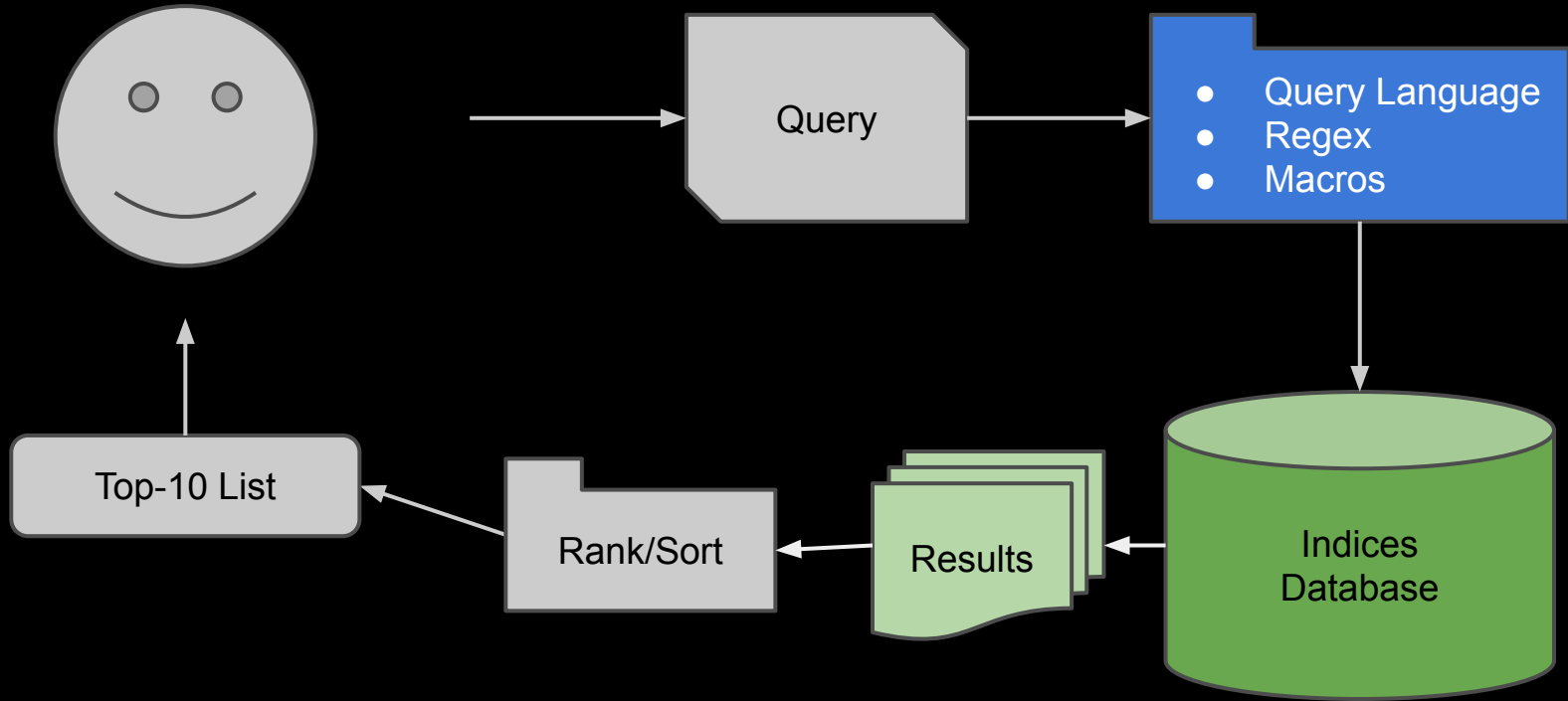
Web search - What's an Index?

Helpful to index other ranking/scoring methods:

- PageRank
- TF-IDF
- Vector space for cosine similarity
- Latent semantic analysis -> LSI



Web search



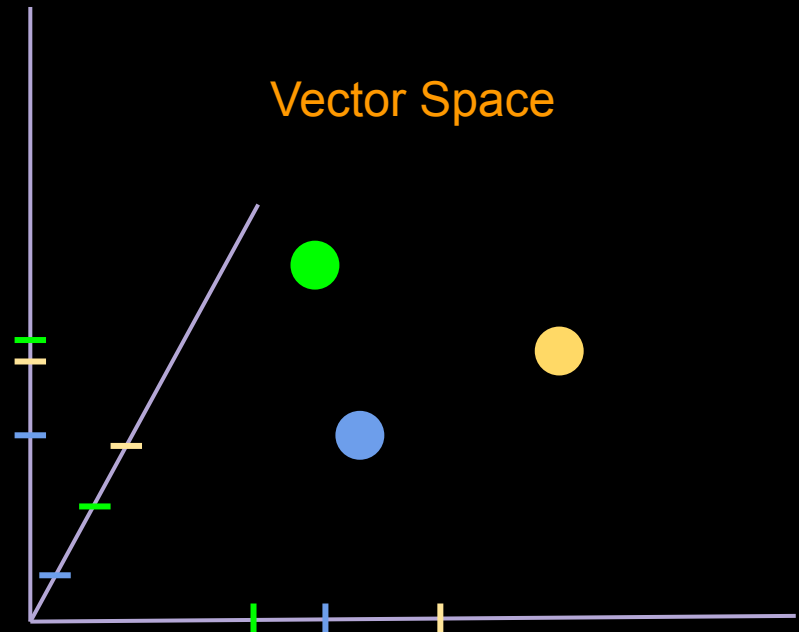
Ranking

$$R(D, Q) = h_1(D, Q) + h_2(D, Q) + h_3(D) \dots$$

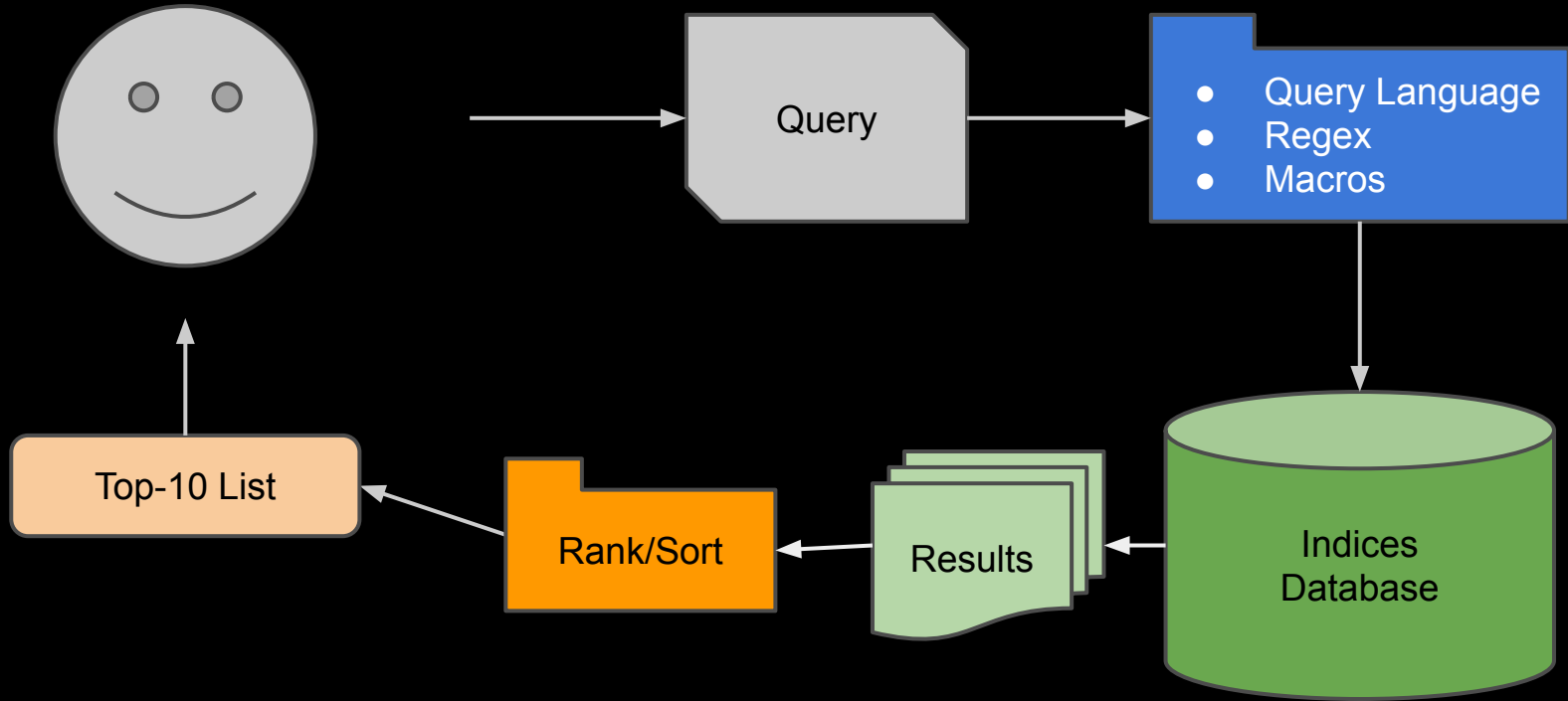
Heuristics:

- TF-IDF
- PageRank

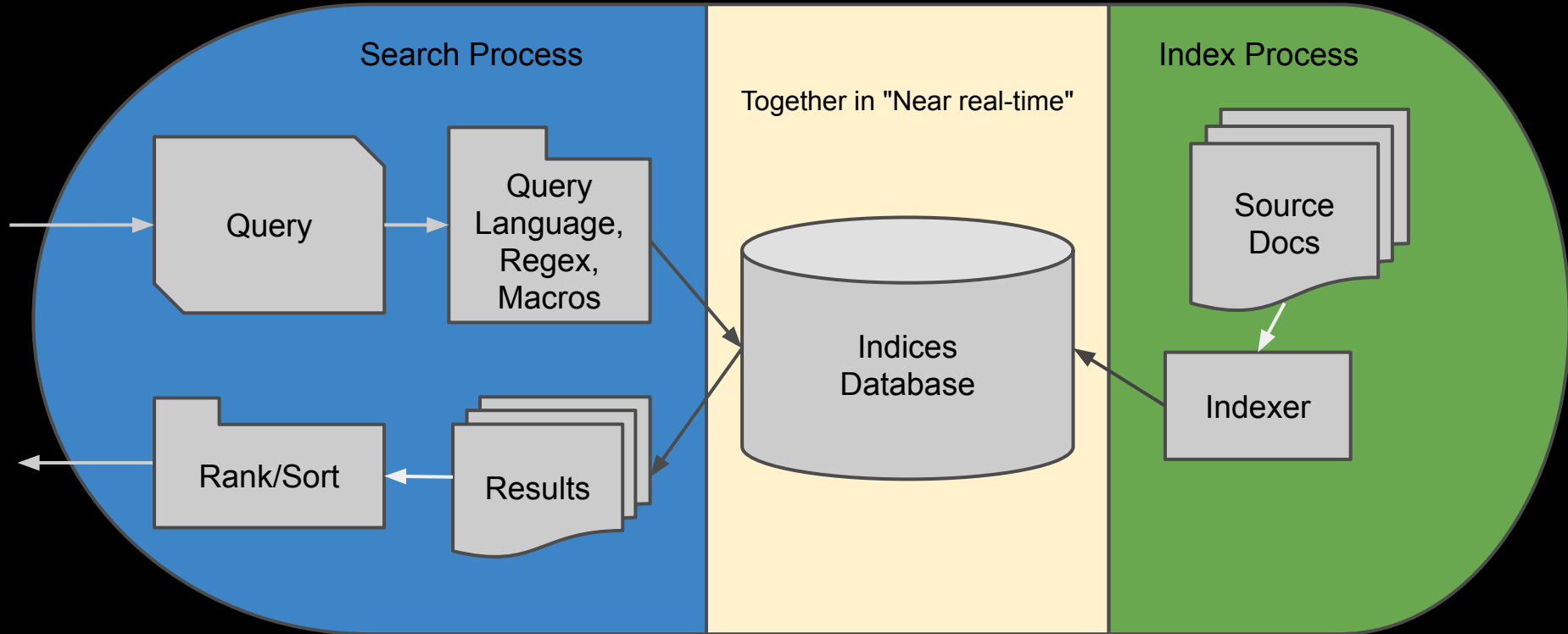
... and return top 10.



Web search



Web search



Intro to *Lucene*

Lucene is the backend. Others build on top.

Elasticsearch / Solr = HTTP interface

Blender = Twitter's proprietary frontend

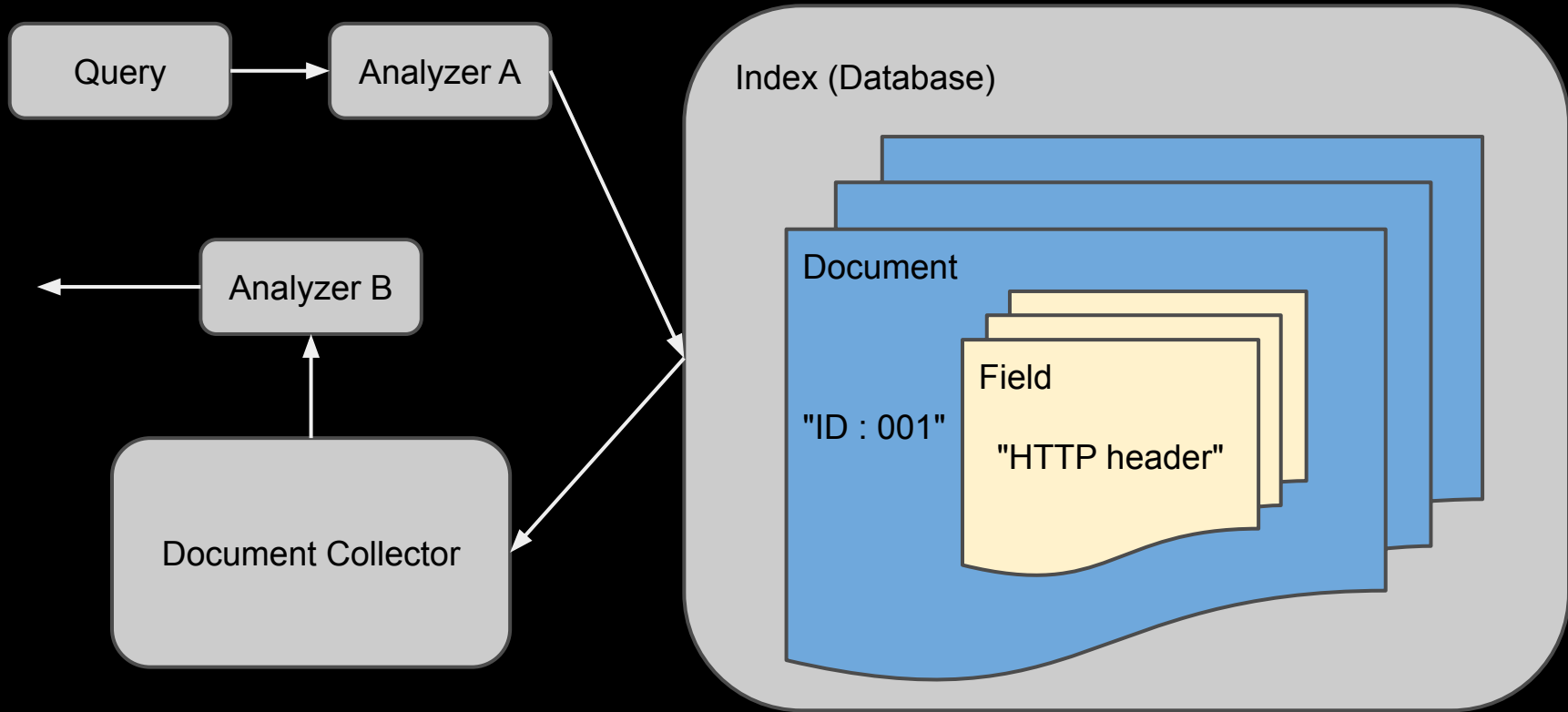
Who uses Lucene

- Twitter
- LinkedIn
- ...

Who uses Solr

- Netflix
- eBay
- ...

Lucene model



Lucene code example - Indexing

```
public class OurIndexer
{
    public static void main( String[] args ) throws IOException
    {
        OurIndexer indexer = new OurIndexer();
        indexer.makeIDMaps();
        indexer.indexFileOrDirectory("../run2/crawl/text");
        indexer.closeIndex();
    }
}
```

Lucene code example - Indexing

```
public class OurIndexer
{
    public OurIndexer() throws IOException
    {
        StandardAnalyzer analyzer = ...;    // Preprocessing
        IndexWriterConfig config =
            new IndexWriterConfig(analyzer);
        writer = new IndexWriter(dir, config);
    }
}
```

Lucene code example - Indexing

```
public void indexFileOrDirectory(String resourcePath) throws IOException
```

```
{
```

```
    ...  
    Document doc = new Document(); fr = new FileReader(f);  
    String docid = ... ; String url = id2url.get(docid); String title = id2title.get(docid);
```

```
    String[] spliturl = url.split("/"); // Heuristic for ranking boost.  
    float boost = (1 / (float) spliturl.length);
```

```
    TextField content_field = new TextField("contents", fr); // Tokenize or Stem?  
    content_field.setBoost(boost);  
    doc.add(content_field);
```

```
    StringField idfield = ...; doc.add(idfield);  
    StringField urlfield = ...; doc.add(urlfield);  
    TextField titlefield = ...; doc.add(titlefield);
```

```
    writer.addDocument(doc);
```

```
}
```

Lucene code example - Searching

```
private static void simpleSearch(String queryString) throws ParseException, IOException
{
    TopScoreDocCollector collector = TopScoreDocCollector.create(RESULTS_LENGTH, true);

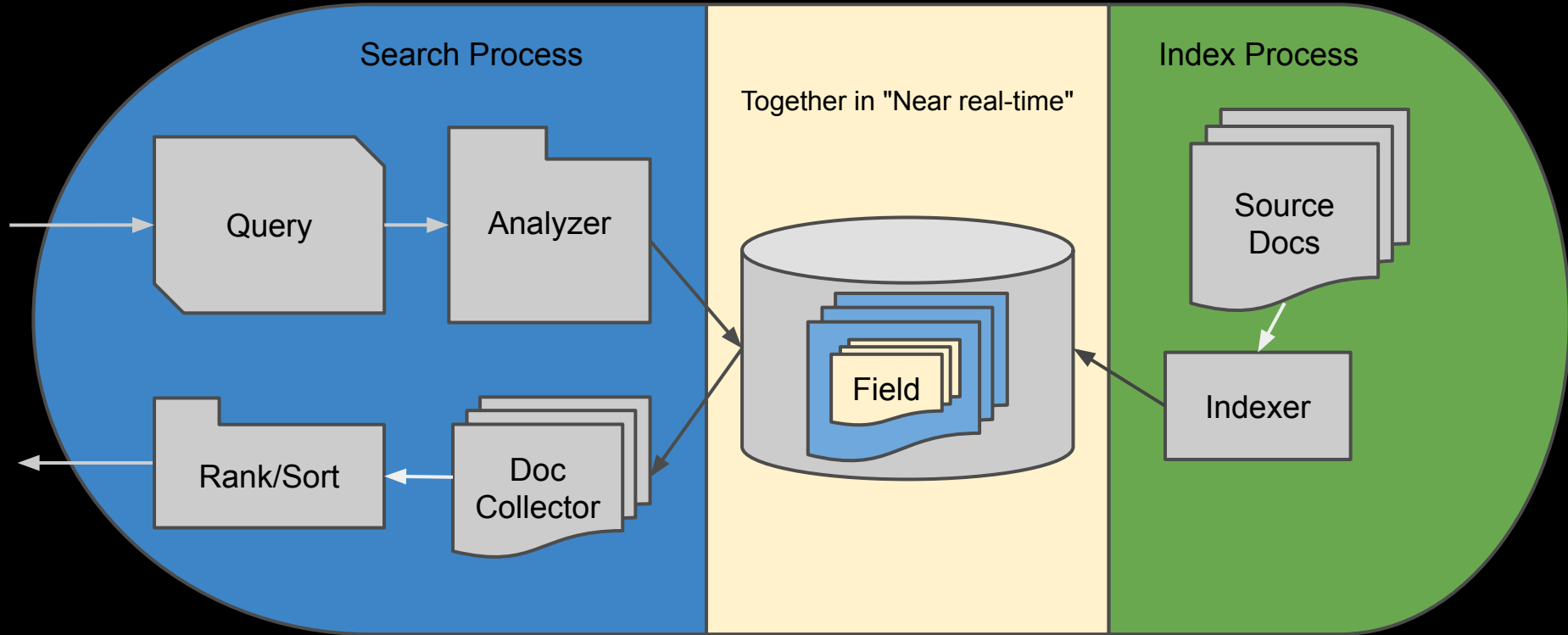
    Query query = new QueryParser(DOC_CONTENT_FIELD, analyzer ).parse(queryString);

    searcher.search(query, collector);

    ScoreDoc[] hits = collector.topDocs().scoreDocs;

    // return or print results
    ...
}
```

You've done Lucene



The Plan

- Web search engines
 - Aspects of text
 - Ranking
 - Setting up Lucene
- Non-text data

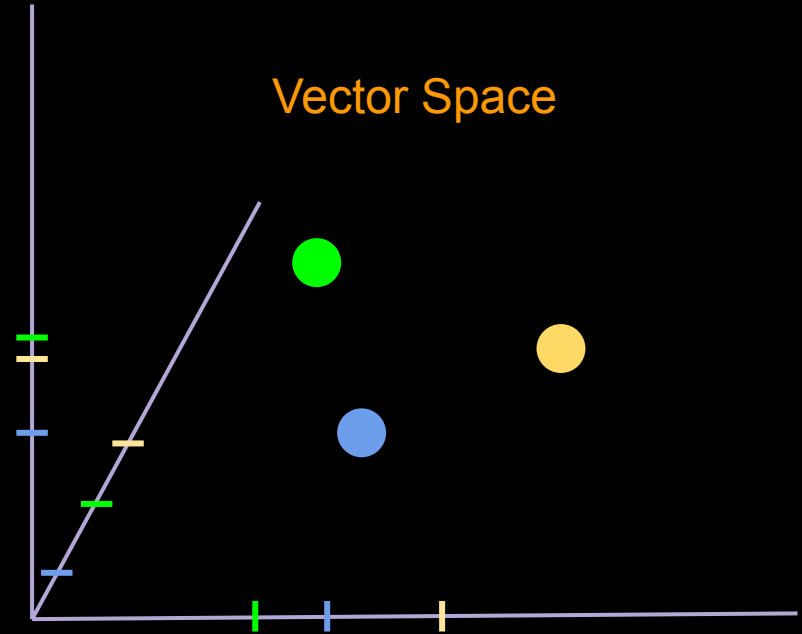
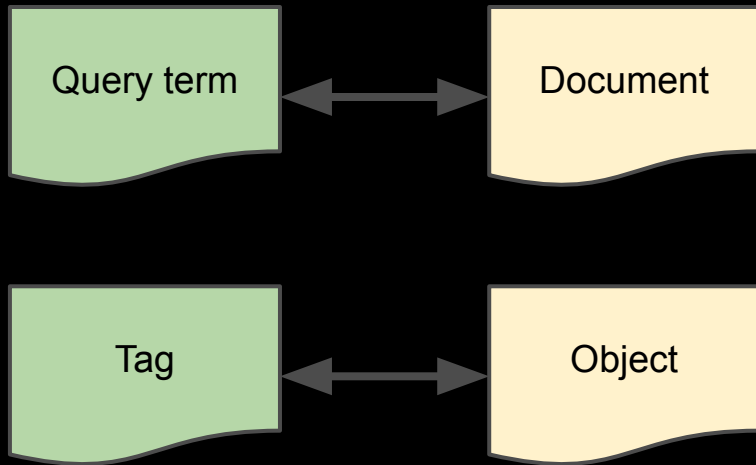
time permitting:

- Alternate search paradigms
 - User-driven exploration

Not covered

- sharding
- hadoop
- advanced text processing

What about searching _____ ?

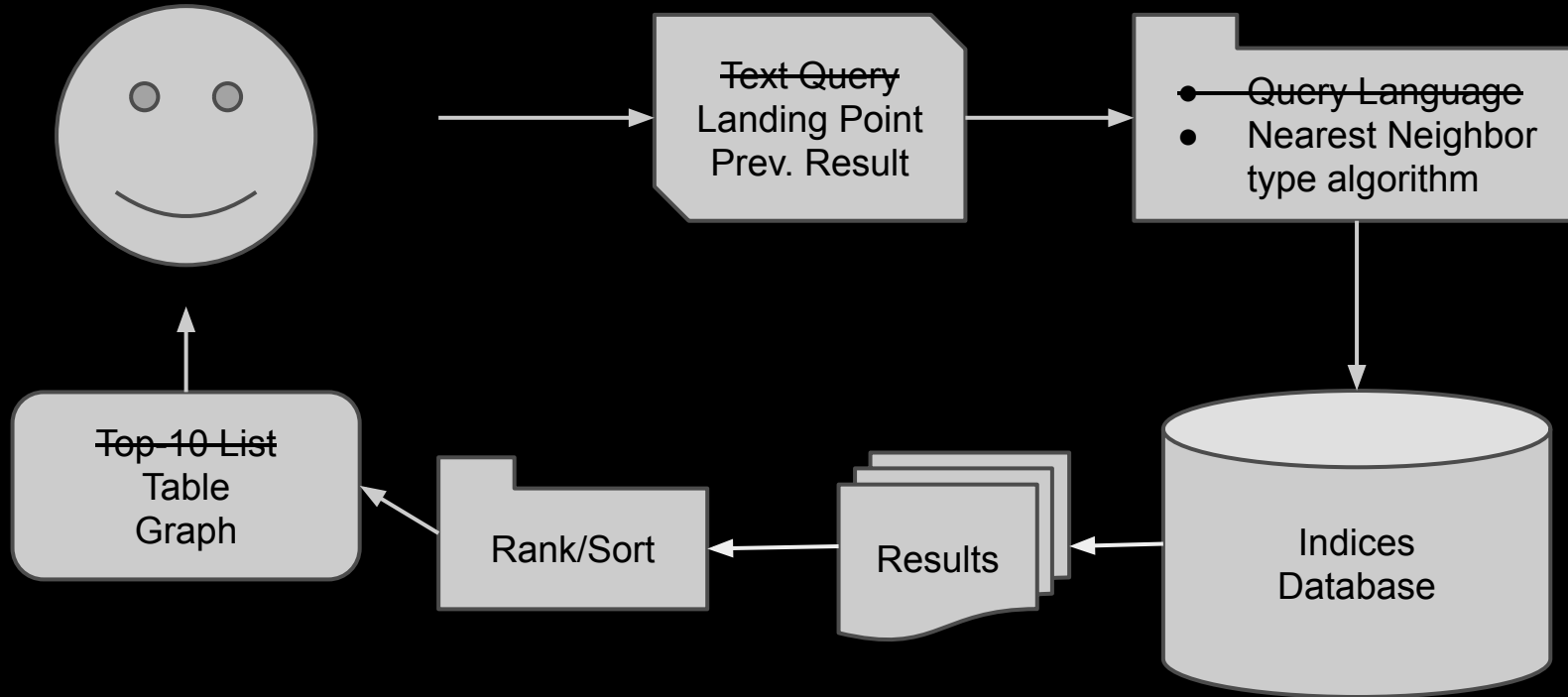


What about searching _____ ?

Extend this model using vector-space indexing.

	Doc-1 Object 1	Doc-2 Object 2	Doc-3 Object 3
Word-1 Tag 1	5	8	8
Word-2 Tag 2	2	0	6
Word-3 Tag 3	1	7	0

Search as Network Exploration



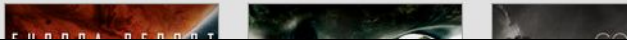
Searching movies?

Science Fiction

Explore titles related to

Titles related to Science Fiction

Science Fiction



Starship Troopers

1997 R 2hr 9m

★★★★★

Our best guess for Michael: 4.8 stars

Average of 4,142,299 ratings: 3.5 stars

Genres

- Action Thrillers
- Sci-Fi Thrillers
- Sci-Fi & Fantasy
- Alien Sci-Fi
- Action & Adventure
- Cult Sci-Fi & Fantasy
- Cult Movies
- Action Sci-Fi & Fantasy
- Monster Movies

This movie is

- Violent
- Imaginative

Exploring movies

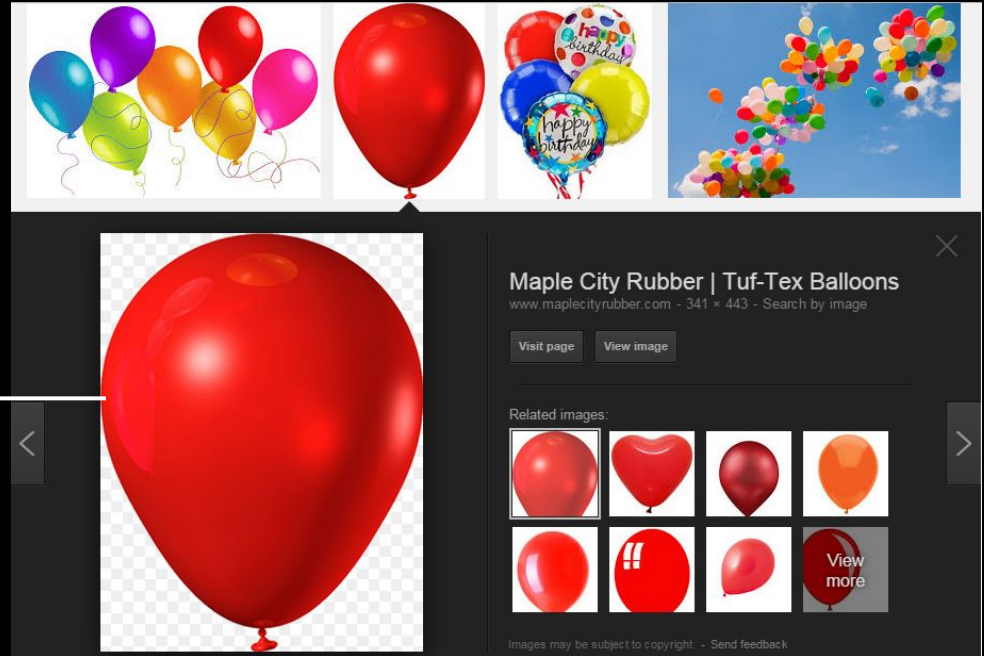
Top Picks for Michael

More Like *Starship Troopers*




Searching pictures?

	Doc-1 Link 1	Doc-2 Link 2
Word-1 URL	5	8
Word-2 Filename	2	0
Word-3 Tag 1	1	7
...		



Searching network profiles?

	Profile 1	Profile 2
em algorithm	191	...
machine learning	113	...
computational biology	191	...
graphical models	80	



Michael I. Jordan Follow

Professor of EECS and Professor of Statistics, University of California, Berkeley
machine learning, statistics, computational biology, artificial intelligence
Verified email at cs.berkeley.edu - Homepage

em algorithm

Scholars that used it: 242
Instances : 6388

Scholars	Frequency of the concept under the scholar
nikos vlassis	196
michael i jordan	191
zoubin ghahramani	174
robert d nowak	133
nando de freitas	101
chris williams	99
cédric archambeau	98
maya gupta	96
vair weiss	87

Where to get tags?

By hand

- crowdsourcing
 - #hashtag
 - mechanical turk
- technician logging

ML text processing

- Neural Nets for Image Captioning
- Text Segmentation
- Summarization

The End

- Web search engines
 - Aspects of text
 - Ranking
 - Setting up Lucene
- Non-text data

time permitting:

- Alternate search paradigms
 - User-driven exploration

Not covered

- sharding
- hadoop
- advanced text processing

Recommended References

[RegEx Engine Mechanics](#)

[Dealing with Malicious Bots/Crawlers](#)

[The Engineering Behind Twitter's New Search](#)

[Michael's Notes](#)