https://vvtesh.sarahah.com/

Information Retrieval

Venkatesh Vinayakarao

Term: Aug – Dec, 2018
Indian Institute of Information Technology, Sri City

So much of life, it seems to me, is determined by pure randomness.
– Sidney Poitier.

Big Data

Hadoop, Map Reduce and Pig

Near Duplicates

- How to detect near duplicate documents?

Near Duplicates

- How to detect near duplicate documents?

Document d1
Tendulkar hits a century.

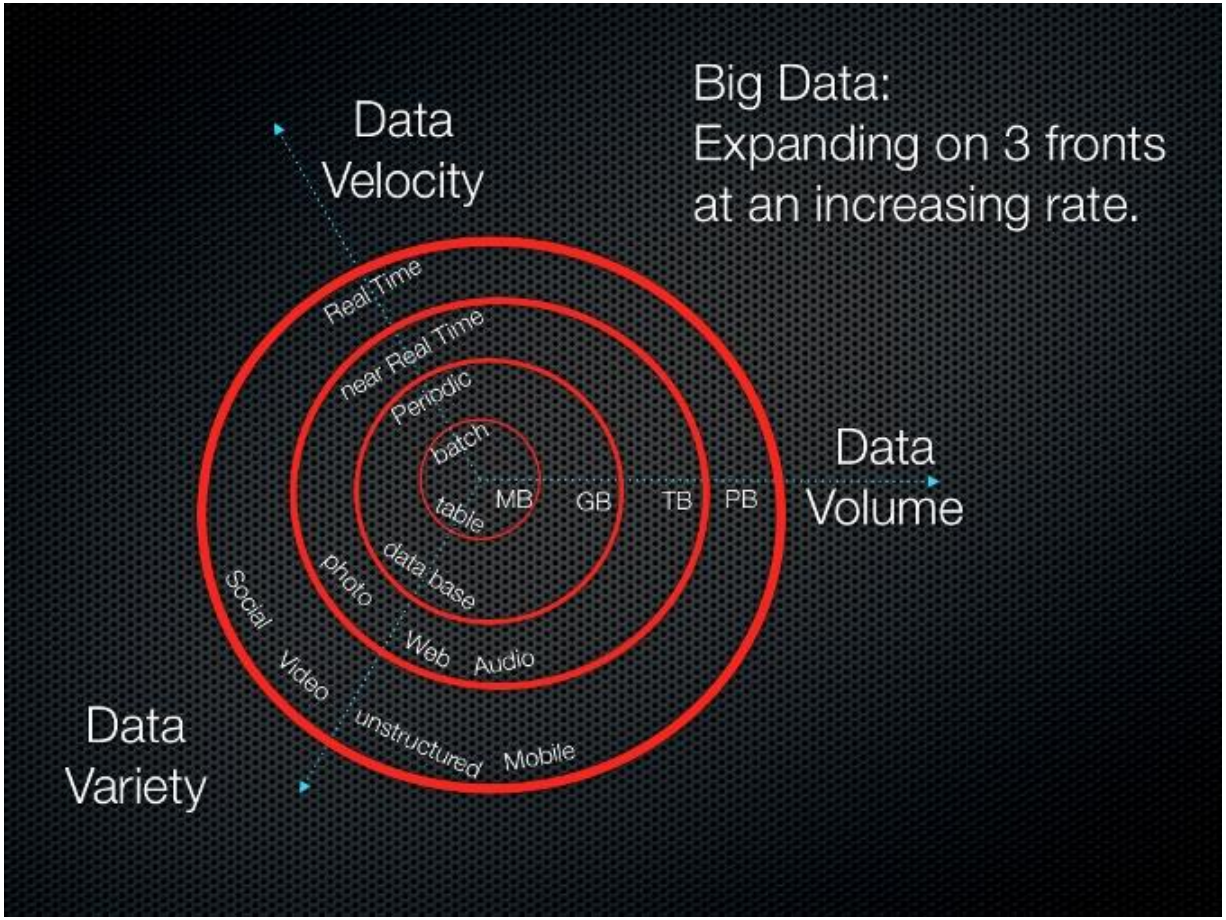
Document d2
Tendulkar hits a fantastic century.

k-Shingles
{Tendulkar hits,
hits a,
a century }

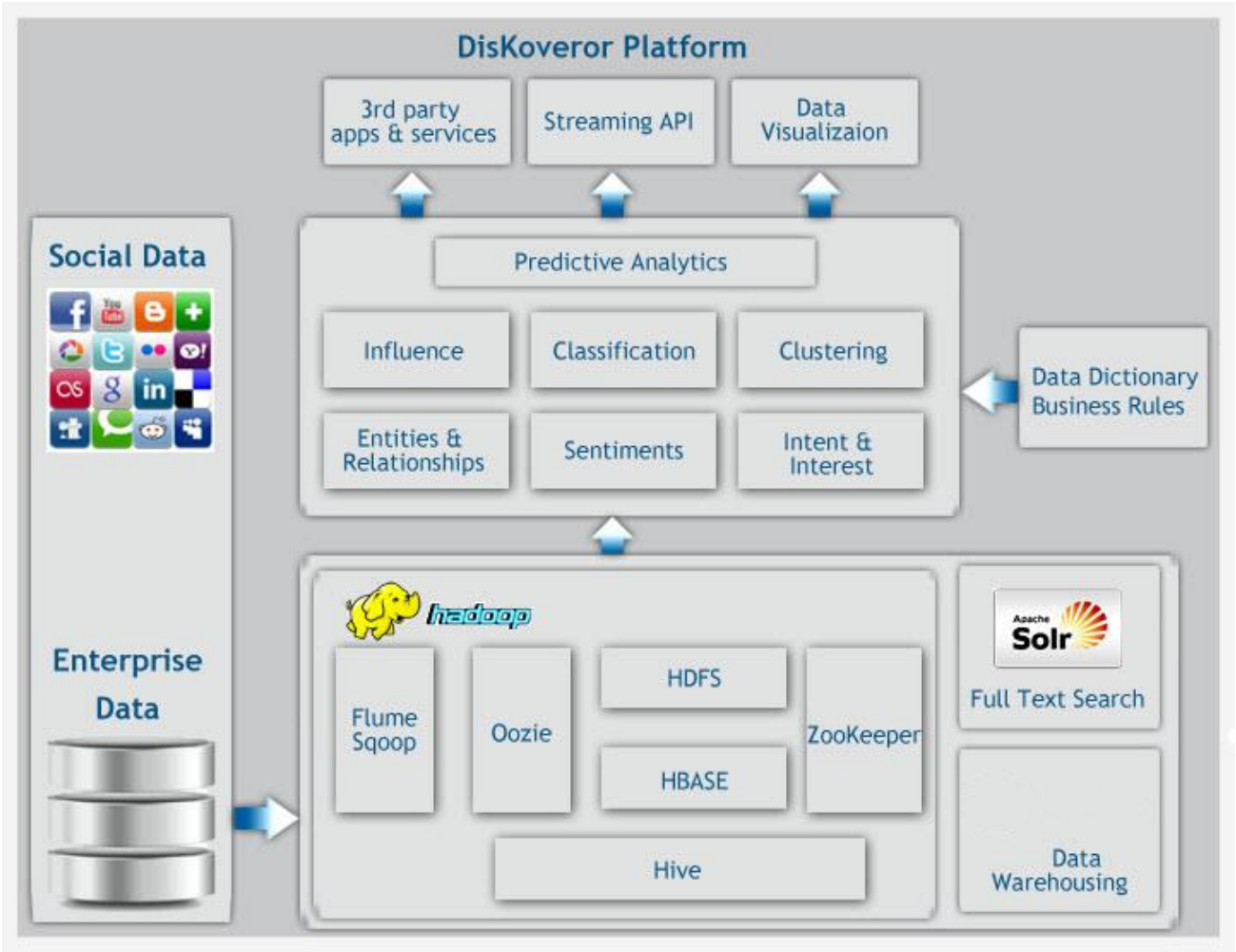
k-Shingles
{Tendulkar hits,
hits a,
a fantastic,
fantastic century}

Apply Jaccard Similarity $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$

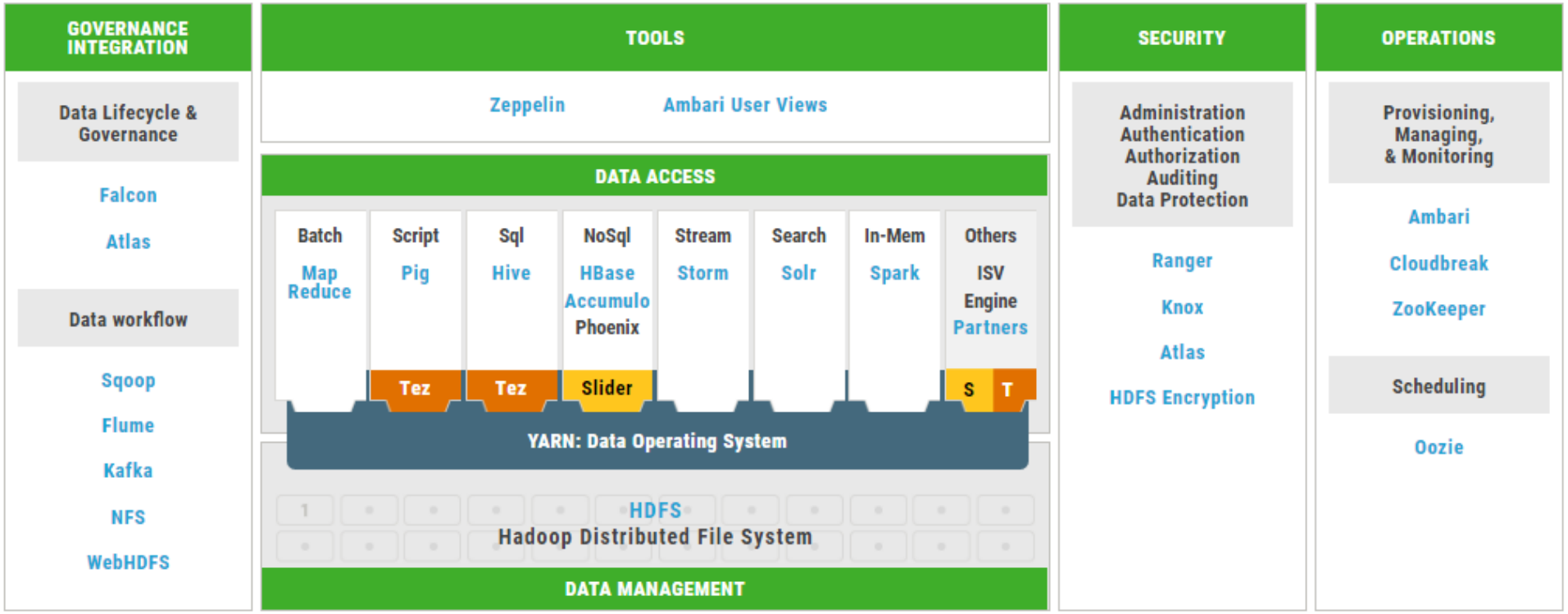
3Vs of Big Data



An Example of Distributed Platforms

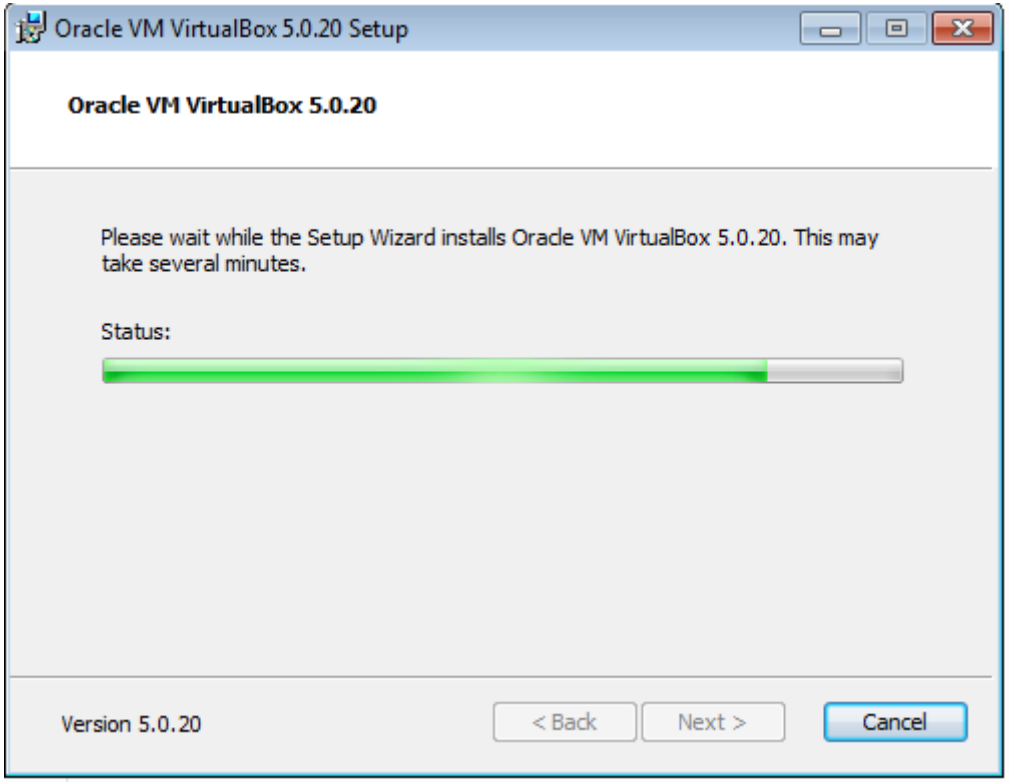


HDP: Hortonworks Data Platform

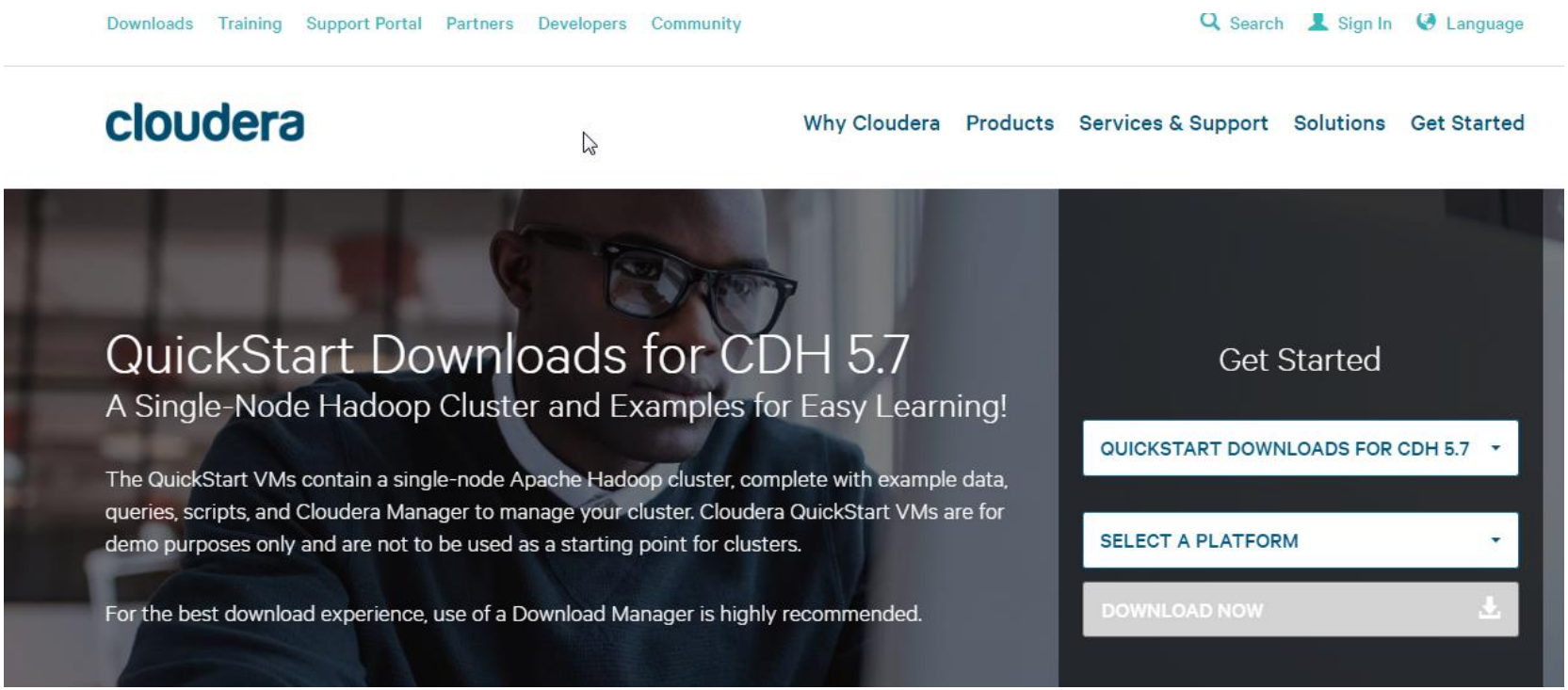


Hadoop Installation

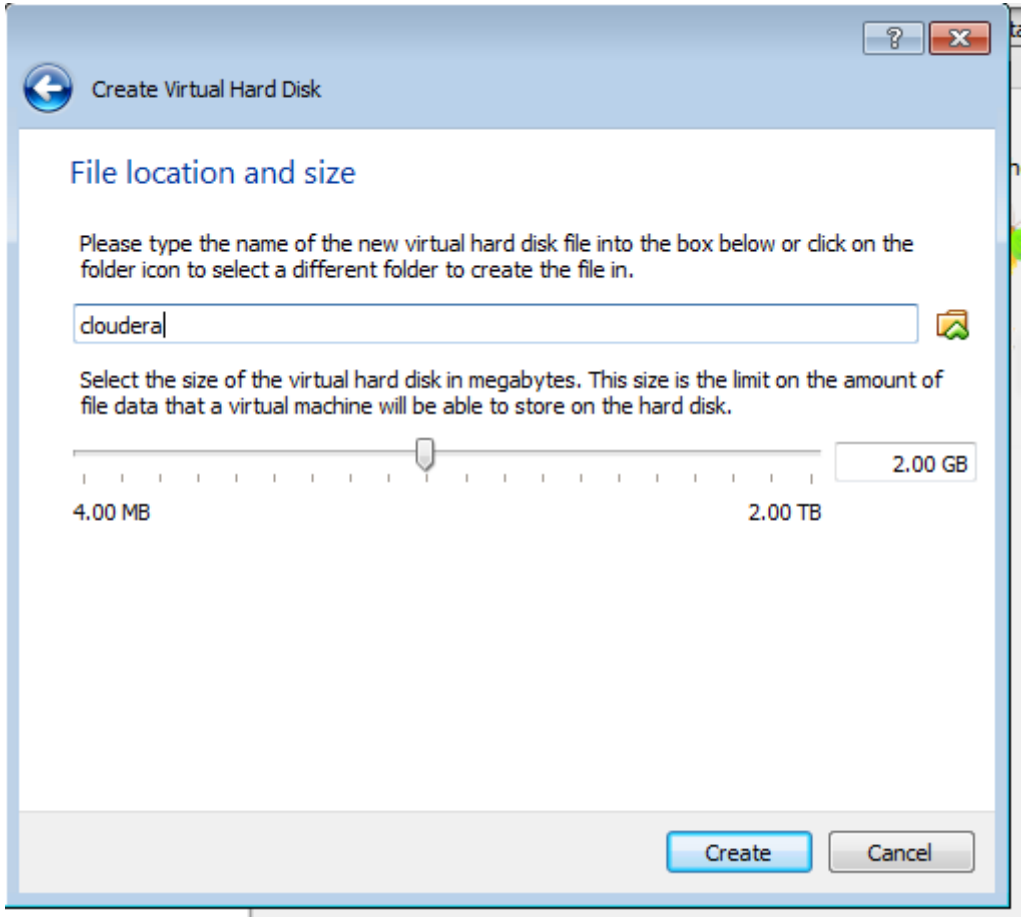
Virtualbox Installation



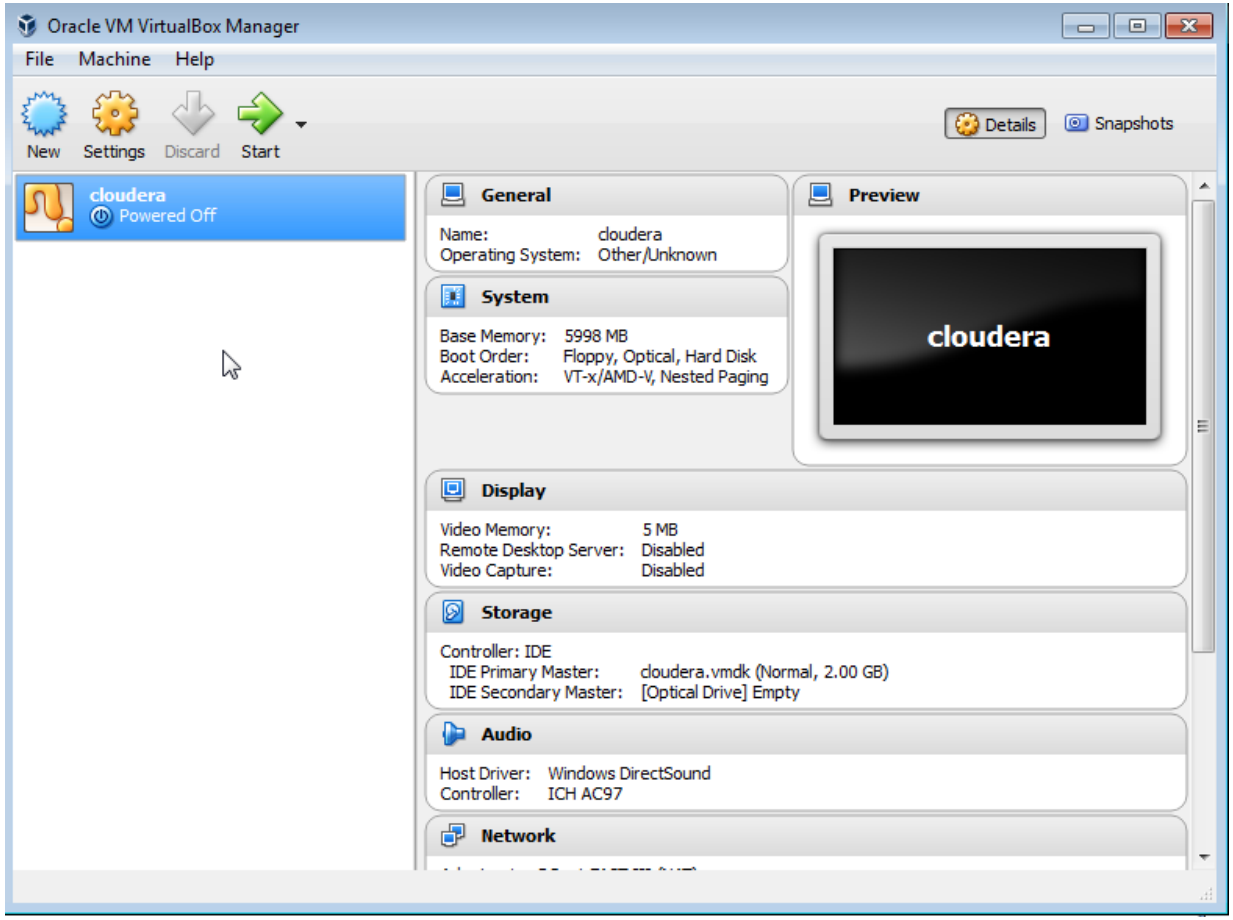
Cloudera VM installation



Virtualbox setup

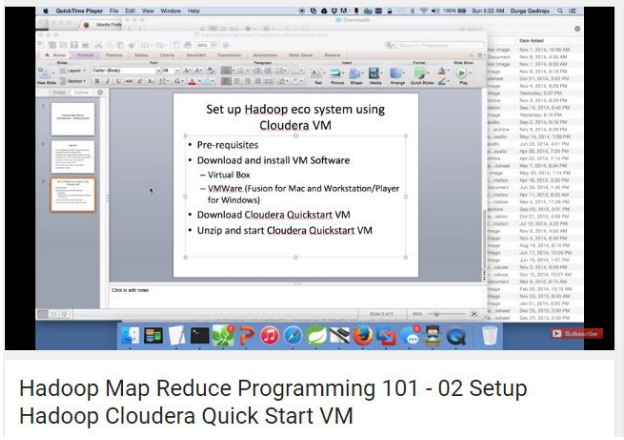


Adding cloudera vm



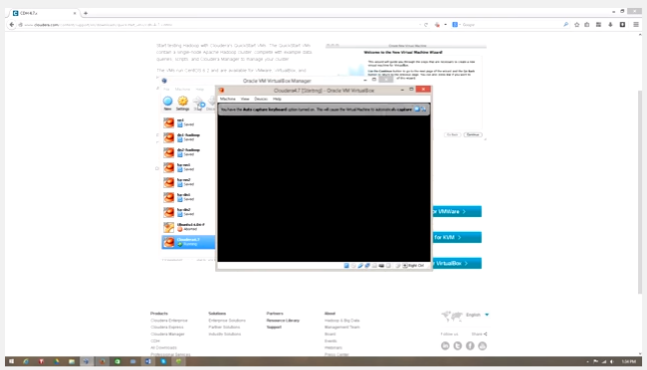
Installation Video

https://www.youtube.com/watch?v=p7uCyFfWL-c



Hadoop Map Reduce Programming 101 - 02 Setup Hadoop Cloudera Quick Start VM

https://www.youtube.com/watch?v=L0lPPC5qeyU



Installing Cloudera VM on Virtualbox on Windows

BigData 101

Advanced Map Reduce: Job Chaining

- How to find top k words?
  - Write one MR job for word count.
  - Write another MR job to process the output to find top k.
    - Sometimes, the output of job 1 is not so big. You do not need MR job to do the task. You may just use shell script or manually do it.
    - For example,

```
hadoop fs -cat /output/of/wordcount/part* | sort -n -k2 -r | head -n20
```

Reduce - Optimization

- How to do top k?
  - Have each reducer only output k results.
  - If you have only one reducer, this problem is already solved.

Example

- https://github.com/adamjshook/mapreducepatterns/blob/master/MRDP/src/main/java/mrdp/ch3/TopTenDriver.java

```
public void reduce(NullWritable key, Iterable<Text> values,
                Context context) throws IOException, InterruptedException {
        for (Text value : values) {
                Map<String, String> parsed = MRDPUtils.transformXmlToMap(value
                                .toString());

                repToRecordMap.put(Integer.parseInt(parsed.get("Reputation")),
                                new Text(value));

                if (repToRecordMap.size() > 10) {
                        repToRecordMap.remove(repToRecordMap.firstKey());
                }
        }

        for (Text t : repToRecordMap.descendingMap().values()) {
                context.write(NullWritable.get(), t);
        }
}
```

MR Design Patterns

Taxonomy of Patterns

1. Summarization
    1. Average
    2. Median
2. Filtering
    1. Top k
    2. Distinct
3. Joins
    1. Cartesian Product
4. Meta patters
    1. Job Chaining
5. Data Organization
    1. Partitioning
    2. Bin'ning
6. External I/O

Pig

A scripting language to simplify processing on Hadoop.

```
A = load 'passwd' using PigStorage(':');  -- load the passwd file
B = foreach A generate $0 as id;  -- extract the user IDs
store B into 'id.out';  -- write the results to a file name id.out
```

```
$ pig -x local id.pig
```

More information at http://pig.apache.org/docs/r0.14.0/start.html

Benefits & Limitations

- Benefits
  - 10 lines of Pig Latin (approx.) = 200 lines in Java
  - 15 minutes in Pig Latin (approx.) = 3 hours in Java
    - Simple
    - Easy
    - Quick to Code
- Limitations
  - Slower than Map-Reduce

Word Count

```
Lines=LOAD ‘input/hadoop.log’ AS (line: chararray);
Words = FOREACH Lines GENERATE
    FLATTEN(TOKENIZE(line)) AS word;
Groups = GROUP Words BY word;
Counts = FOREACH Groups GENERATE  group,
    COUNT(Words);
Results = ORDER Words BY Counts DESC;
Top5 = LIMIT Results 5;
STORE Top5 INTO /output/top5words;
```

Pig in Real-World

- Yahoo used it extensively (>70% of jobs)
- Facebook – Process Logs
- Twitter – Process Logs
- eBay – Data processing for intelligence
- …

UDF in Pig

```
-- myscript.pig
REGISTER myudfs.jar;
A = LOAD 'student_data' AS (name: chararray, age: int, gpa: float);
B = FOREACH A GENERATE myudfs.UPPER(name);
DUMP B;
```

Simple UDF

```
…
6  public class UPPER extends EvalFunc<String>
7  {
8    public String exec(Tuple input) throws IOException {
9       if (input == null || input.size() == 0)
10            return null;
11        try{
12            String str = (String)input.get(0);
13           return str.toUpperCase();
14        }catch(Exception e){
15            throw new IOException("Caught exception processing input row ", e);
16        }
17    }
18  }
```

Source: https://pig.apache.org/docs/r0.10.0/udf.html

Creating the Jar

Javac -cp /..path../pig.jar UPPER.java

jar -cf exampleudf.jar exampleudf

Know where have you placed this jar.

In Pig Script:

- REGISTER '…path to jar';
- DEFINE SIMPLEUPPER exampleudf.UPPER();
- … now you can use this method.

https://vvtesh.sarahah.com/

Information Retrieval

Venkatesh Vinayakarao

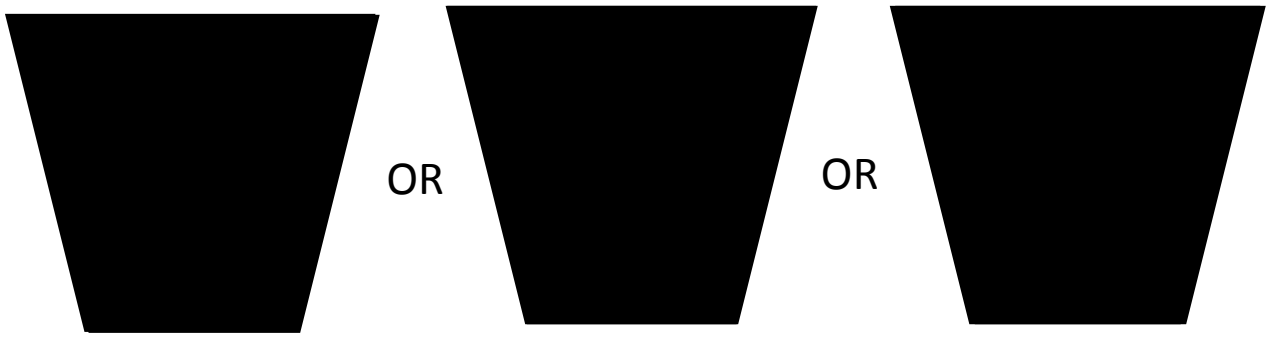
Term: Aug – Dec, 2018
Indian Institute of Information Technology, Sri City

So much of life, it seems to me, is determined by pure randomness.
– Sidney Poitier.

The Intuition

- Which bucket is most likely to lead to a

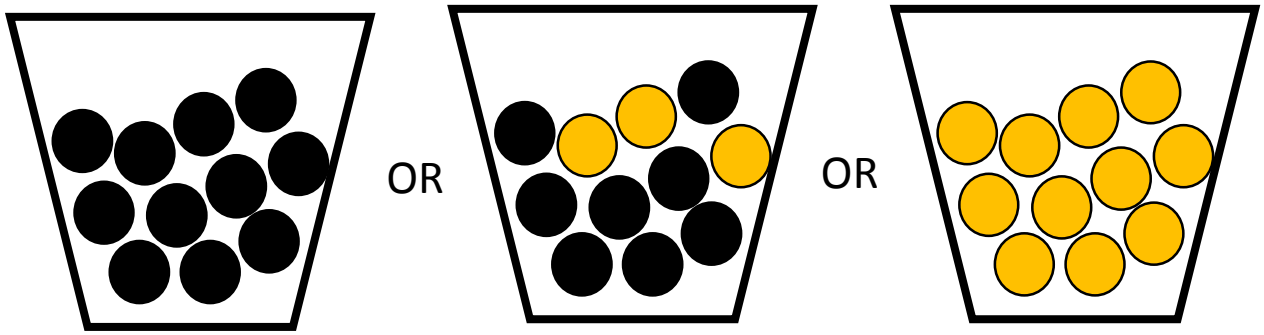
Black Ball?



The Intuition

- Which bucket is most likely to lead to a

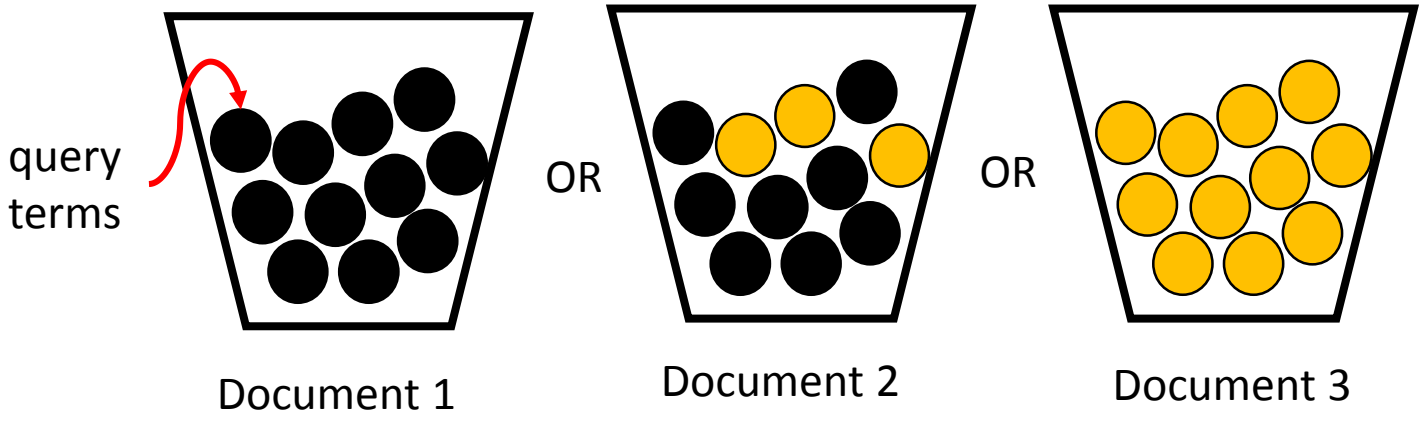
Black Ball?



The Intuition

- Which document is most likely to lead to a

query?



Can be modeled using
a discriminative model (P(D|R=1,Q)) or a generative model (P(Q|R=1,D))

A good query…

Words that are most likely to appear in a relevant document

Another Way to Rank

Rank documents based on the probability of the model generating the query: $P(q|M_d)$

M_d is the model of the document which generates the query

A Model to Generate a String

- What strings can this model generate?

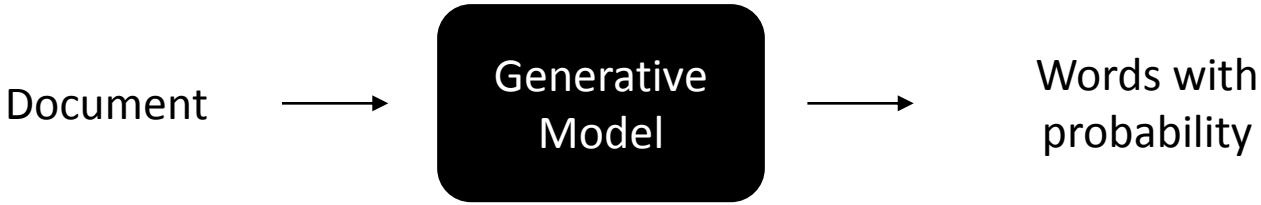
This is a Finite Automaton that can generate:

I wish

I wish I wish

I wish I wish I wish …

Key Idea



Example

- P(STOP|t) = 0.2 i.e., probability that our automaton stops after encountering any word is 0.2.
- You are given with the probabilities of words appearing in the query.
- What is probability of “frog said that toad likes frog” being the query?
  - Answer: (0.01 x 0.03 x 0.04 x 0.01 x 0.02 x 0.01) x (0.8^5 x 0.2) = 0.000000000001573

Word	Probability
the	0.2
a	0.1
frog	0.01
toad	0.01
said	0.03
likes	0.02
that	0.04

Uday: It is much easier to take log and add instead of multiplying these numbers.

Comparing Models

- Say we have two models:

s	frog	said	that	toad	likes	that	dog
M_1	0.01	0.03	0.04	0.01	0.02	0.04	0.005
M_2	0.0002	0.03	0.04	0.0001	0.04	0.04	0.01

$P(s|M_1) = 0.00000000000048$
$P(s|M_2) = 0.000000000000000384$

- Model 1 seems to have higher probability of generating the string.
- Model 1 will match a document containing these terms better to this query.

A Simple Language Model

- Simple Model

$$P(t_1 t_2 t_3 t_4) = P(t_1)P(t_2|t_1)P(t_3|t_1 t_2)P(t_4|t_1 t_2 t_3)$$

- Even simpler – The Unigram Language Model

$$P_{\text{uni}}(t_1 t_2 t_3 t_4) = P(t_1)P(t_2)P(t_3)P(t_4)$$

- The Bigram Model

$$P_{\text{bi}}(t_1 t_2 t_3 t_4) = P(t_1)P(t_2|t_1)P(t_3|t_2)P(t_4|t_3)$$

Document as a Bag of Words

- We rank documents based on the probability of that document to generate the given query.

$$P(d|q) = P(q|d)P(d)/P(q).$$

- If documents are bag of words,

$$P(q|M_d) = K_q \prod_{t \in V} P(t|M_d)^{\mathrm{tf}_{t,d}}$$

- Assume p(d) is uniform. Hence can be dropped.
- P(q) and K_q does not affect ranking. Hence can be dropped.

Missing Terms

- What happens to P(d|q) if a query term does not appear in the document?

Clue: We approximate P(d|q) to document ranking by computing $\prod_{t \in V} P(t|M_d)^{\mathrm{tf}_{t,d}}$

Missing Terms & Smoothening

- Instead of zero, use $P(t|M_c)$
- M_c is the model over the entire collection.
- **Smoothen**: We can also mix document and collection probabilities using a linear interpolation co-efficient λ.

$$P(d|q) \propto P(d) \prod_{t \in q} ((1-\lambda) P(t|M_c) + \lambda P(t|M_d))$$

Wake Up Quiz

- Is the following same as the document length?
  - True or False?

$$\sum_{1 \le i \le M} tf(ti, d)$$

M is the term vocabulary size. Function tf(t,d) is term frequency of a term in a document.

Example

- Suppose:
  - d1: Tada is a city between Chennai and Tirupati
  - d2: Tada has few restaurants but no good malls
- Use the smoothened MLE unigram language model to rank these documents for the query “Tada City”. Assume $\lambda = 0.5$.

Example

$$\prod_{t \in q} ((1-\lambda) P(t|M_c) + \lambda P(t|M_d))$$

- Suppose:
  - d1: Tada is a city between Chennai and Tirupati
  - d2: Tada has few restaurants but no good malls
- Use the smoothened MLE unigram language model to rank these documents for the query “Tada City”. Assume $\lambda = 0.5$.
  - *P(q|d1) = [(1/8 + 2/16)/2] · [(1/8 + 1/16)/2] = 1/8 · 3/32 = 3/256*
  - *P(q|d2) = [(1/8 + 2/16)/2] · [(0/8 + 1/16)/2] = 1/8 · 1/32 = 1/256*
  - *Ranking: d1 > d2*

Summary

- Probabilistic Retrieval

Developers in two companies

	Java	C	Total
Company-X	1	17	18
Company-Y	37	20	57
Total	38	37	75

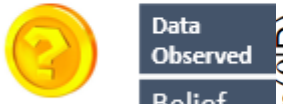
	Jav
Company-X	0.0
Company-Y	0.4
Total	

P(Company-Y|Java)

P(Java|Company-Y)

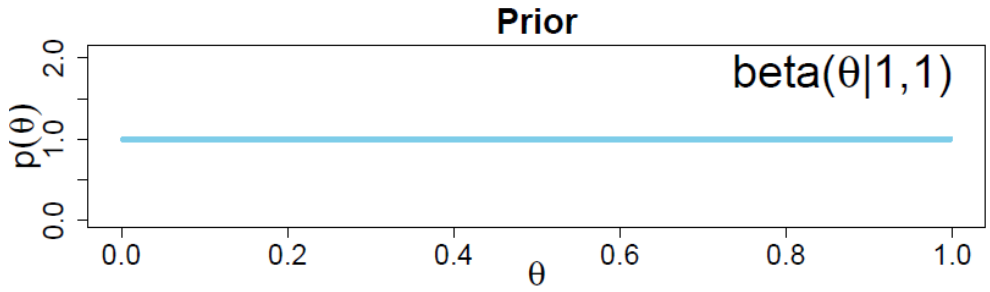
Data Observed

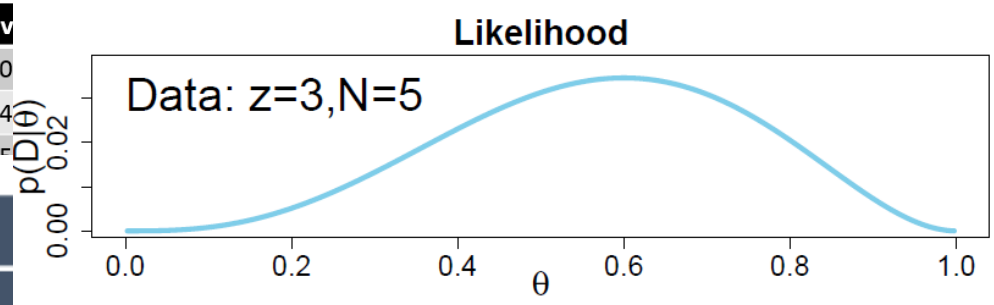
Belief

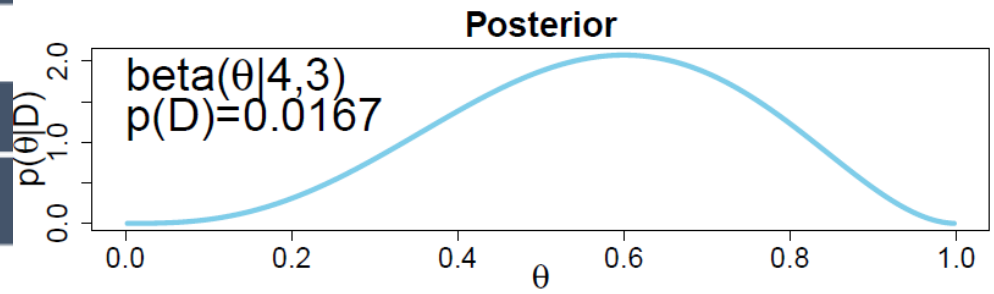


Data Observed

Belief







Prior Belief

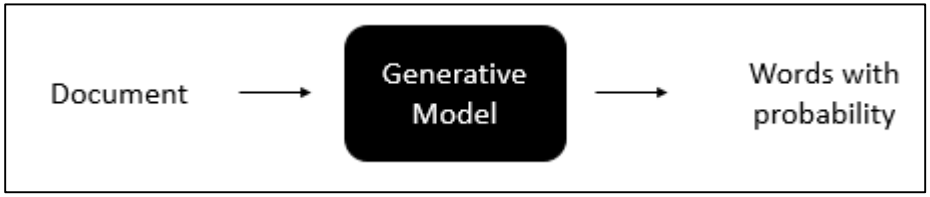
Belief Updates

Summary

- Odds of Relevance

$$\prod_{t:x_t=1} \frac{P(x_t = 1|R = 1, \vec{q})}{P(x_t = 1|R = 0, \vec{q})} \cdot \prod_{t:x_t=0} \frac{P(x_t = 0|R = 1, \vec{q})}{P(x_t = 0|R = 0, \vec{q})}$$

Summary



What is probability of "frog said that toad likes frog" being the query?

- Answer: (0.01 x 0.03 x 0.04 x 0.01 x 0.02 x 0.01) x (0.8^5 x 0.2) = 0.000000000001573

$$P(R = 1|d, q) = \frac{P(d|R = 1, q)P(R = 1|q)}{P(d|q)}$$

Bayes Rule

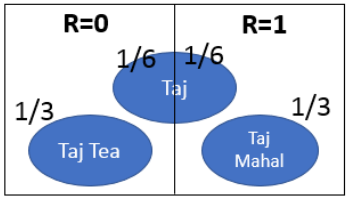
P(d=Taj|R=1,q) = 1/3

P(R=1|q) = 1/2

P(d=Taj|q) = 1/3

P(R=1|d=Taj,q) = (1/3)(1/2)/(1/3) = ½

Document	P(R=0|d,q)	P(R=1|d,q)
Taj	0.5	0.5
Taj Mahal	0	1
Taj Tea	1	0



Summary

- Smoothening

$$\prod_{t \in q} ((1 - \lambda) P(t|M_c) + \lambda P(t|M_d))$$

Thank You