

Randomised Language Modelling

Gambling with Space

David Talbot and Miles Osborne

School of Informatics
University of Edinburgh

ACL 2007 (26th June 2007)



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Language Modelling Challenges

- Good modelling \Rightarrow fewer independence assumptions
 - Higher-order N-grams ($N = 8?$)
- Good estimation requires *much* more data
 - Billions / trillions of words
- Small memory footprint
- Low computational complexity

Language Modelling Challenges

- Good modelling \Rightarrow fewer independence assumptions
 - Higher-order N-grams ($N = 8?$)
- Good estimation requires *much* more data
 - Billions / trillions of words
- Small memory footprint
- Low computational complexity

Language Modelling Challenges

- Good modelling \Rightarrow fewer independence assumptions
 - Higher-order N-grams ($N = 8?$)
- Good estimation requires *much* more data
 - Billions / trillions of words
- Small memory footprint
- Low computational complexity



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Lossless, Quantised and Lossy Models

Lossless Models

Store correct set of N -grams with their exact probabilities

Quantised Models

Store correct set of N -grams with approximate probabilities

Lossy Models

Store an approximation to the correct set of N -grams with approximate probabilities

Information-based Space Lower Bound

Lower Bound on Lossless Representations

$$\log_2 \binom{|\mathcal{U}|}{s} > s(\log_2(|\mathcal{U}|) - \log_2(s)) \text{ bits}$$

are needed to represent s items from a Universe \mathcal{U}

Why

- There are $\binom{|\mathcal{U}|}{s}$ distinct sets of size s in \mathcal{U}
- A distinct code must be assigned to each such set
- $\log_2(x)$ bits are needed to represent x distinct codes

Problem

Any lossless representation scales with $|\mathcal{U}|$ (this is not good)



A Curse of Dimensionality - and Large Corpora

Observed N -grams (s) vs. possible N -grams (\mathcal{U}_N)

	s	$ \mathcal{U}_N = \text{vocab} ^N$	$\log_2 \mathcal{U}_N $
1-gms	0.2M	0.2M	18.09
2-gms	5.4M	78,400M	36.19
3-gms	274M	21,952,000,000M	54.28
4-gms	599M	6,146,560,000,000,000M	72.38
5-gms	842M	1,721,036,800,000,000,000,000M	90.47



All Language Models are Approximate

- Model assumptions are *approximate*
- Not using all available data is *approximate*
- Model reduction - pruning, clustering etc. - is *approximate*
- Parameter estimates from finite data are *approximate*

Bloom filters

Are also approximate but may help us relax the above approximations



All Language Models are Approximate

- Model assumptions are *approximate*
- Not using all available data is *approximate*
- Model reduction - pruning, clustering etc. - is *approximate*
- Parameter estimates from finite data are *approximate*

Bloom filters

Are also approximate but may help us relax the above approximations

Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Representing a Set via Hashing

Problem

Represent a set \mathcal{S} of size n drawn from \mathcal{U} where $n \ll |\mathcal{U}|$

Bloom Filter Solution

Implicitly represent set with m bits and k hash functions

Create

- Hash each item k times setting corresponding bits in m

Query

- Hash a candidate k times, if all bits set report *member* else *non-member*



Using a Bloom Filter

0 0 0 0 0 0 0 0 0 0 0 0 0 0

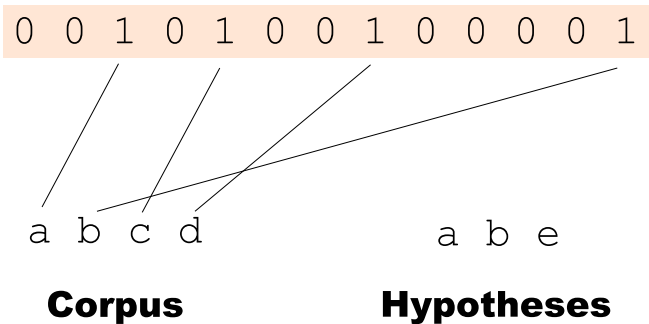
a b c d

Corpus

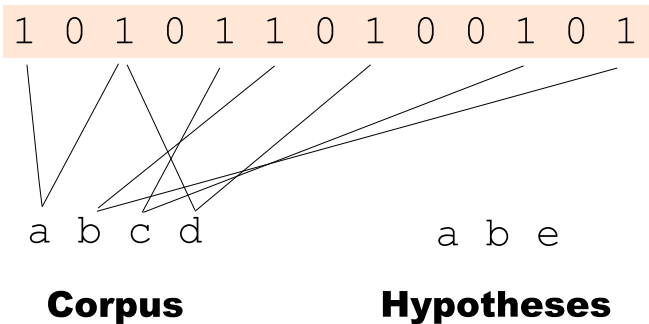
a b e

Hypotheses

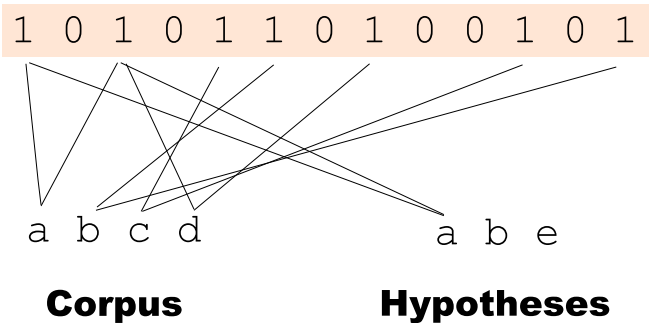
Using a Bloom Filter



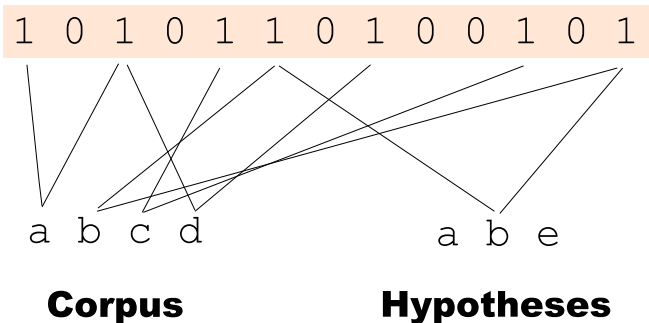
Using a Bloom Filter



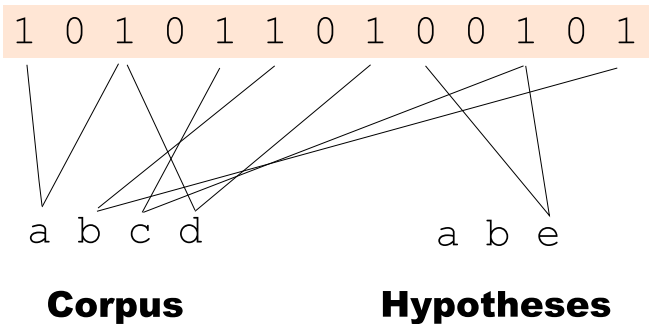
Using a Bloom Filter



Using a Bloom Filter



Using a Bloom Filter



Optimising a Bloom Filter

How many hash functions?

False positive occurs when we observe k bits set *at random*.
False positive rate f is minimised for,

$$k^* = \frac{m}{n} \ln(2).$$

Previous Example: $m = 13, n = 4$

With $k = 1$ the false positive rate was $\frac{4}{13} \approx 0.30$

With $k = 2$ the false positive rate was $(\frac{7}{13})^2 \approx 0.28$

Asymptotically setting half the bits is optimal



Bloom Filter Analysis

Characteristics

- False positives occur with some known (small) probability
- Size and false positive rate *independent* of $|\mathcal{U}|$
- No false negatives - i.e. *one-sided error*

In general for false positive rate $\frac{1}{2^k}$ use

$$\frac{k}{\ln(2)} \text{ bits per object.}$$

Optimality

Bloom Filters are optimal up to this $\frac{1}{\ln(2)} \approx 1.44$ term



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Storing Corpus Statistics

Problem

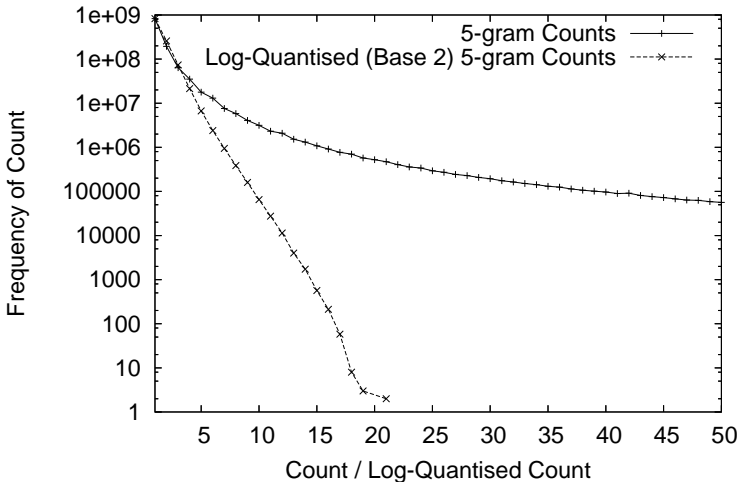
Bloom filters are *not* an associative data structure

Possible Solutions

- 1 Use distinct BF for each count (Bloom Histogram)
 - False positive rate will increase by factor $\approx MAXCOUNT$
 - Computation for a negative will be $MAXCOUNT$
- 2 Replace each bit by a counter (Count-Min Sketch)
 - Space increased by factor $\log(MAXCOUNT)$
 - But most counters will be set to 1 or 2...

Gigaword Corpus Statistics

Distribution of Corpus Statistics



Storing Corpus Statistics

Log Frequency Bloom filter

- Hash each N -gram appended by an integer j

$$1 \leq j \leq 1 + \lfloor \log(count) \rfloor$$

- Query an N -gram's frequency by appending an integer $j = 1$ and incrementing until hitting a 0

Converting Corpus Frequencies to a Set

Raw Counts

the cat	15
the hat	3
the mat	1
the eggs	1
the bacon	1



Quant Counts

4
2
1
1
1

Transformed Set

{the cat_1, the cat_2, the cat_3,
the cat_4, the hat_1, the hat_2,
the mat_1, the eggs_1, the bacon_1}



Log-Frequency BF Analysis (1)

Probability of Overestimation Decays Fast

Errors > 1 correspond to joint occurrence of independent, low probability events.

E.g., with $k = 3$,

$$\Pr\{\hat{q}c(x) = 2 | qc(x) = 1\} = \frac{1}{2^k} = 0.125.$$

but,

$$\Pr\{\hat{q}c(x) = 4 | qc(x) = 1\} = \frac{1}{2^{3k}} = 0.00195.$$

Expected Error

Most of error comes from first erroneous increment hence,

$$E[|qc(x) - \hat{q}c(x)|] \leq \frac{1}{2^k - 2}$$



Log-Frequency BF Analysis (2)

Error is small with high probability

Markov's inequality implies that

$$\Pr\{|qc(x) - \hat{q}c(x)| > \lambda\} \leq \frac{E[err]}{\lambda}.$$

Space Requirements

Log-quantised corpus statistics follow a geometric distribution, hence the average count is < 2

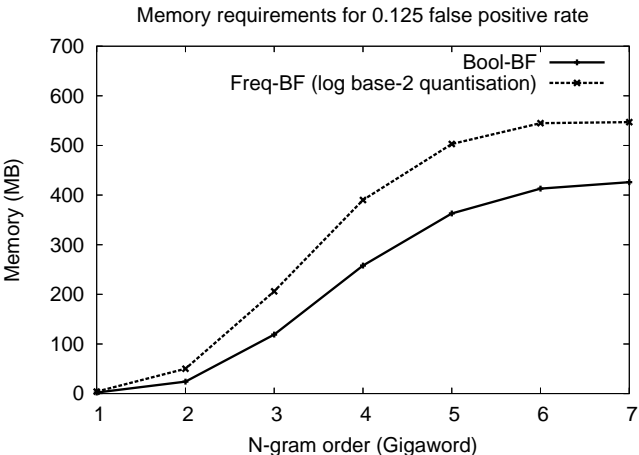
$$\frac{2k}{\ln(2)} < 3k \text{bits per } n\text{-gram}$$

for an expected error of less than $\frac{1}{2^{k-2}}$



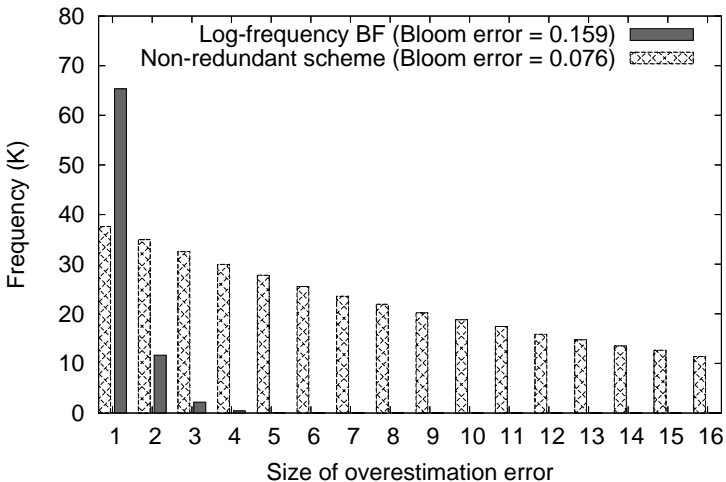
Log Frequency Scheme for Corpus Statistics

- Set increases by *less than 2* when storing frequencies



Log Frequency BF vs. Bloom Histogram

Frequency Estimation Errors on 500K Negatives



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

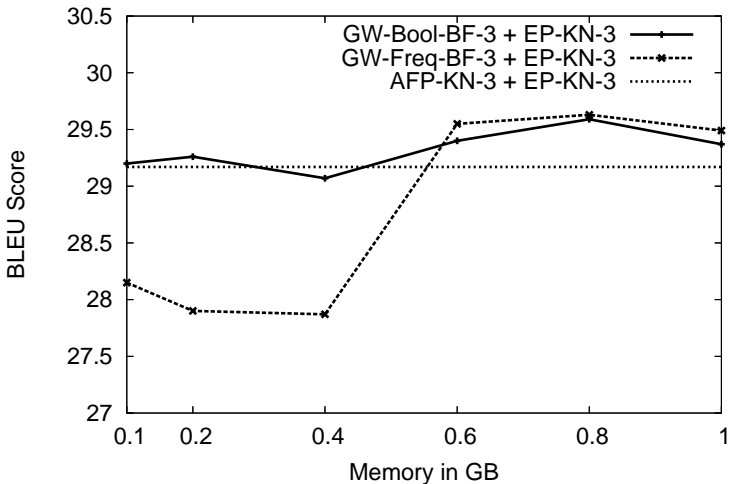
3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



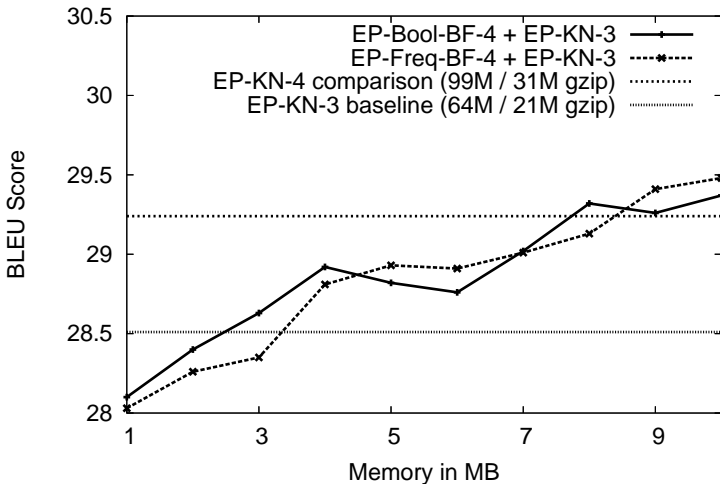
Adding Additional Corpus Statistics with a BF

GW-Bool-BF-3 and GW-Freq-BF-3 (with EP-KN-3)



Adding Higher-Order N -grams with a BF

EP-Bool-BF-4 and Freq-BF-4 (with EP-KN-3)



Outline

1 Motivation

- Scaling Language Modelling
- Lossless vs. Lossy Representations

2 Using Bloom Filters for Language Modelling

- The Bloom Filter
- Extending the BF for Counting Corpus Statistics

3 MT Experiments

- Unsmoothed BF-LM Experiments
- Smoothed BF-LM Experiments



Storing Related Events

Language Model Statistics

- Witten-Bell: N -gram and suffix counts
- Kneser-Ney: N -gram, prefix, suffix and infix counts

Proxy Events

- Use existence of one event to *infer* a related event
e.g. presence of $N - 1$ -gram implies suffix count ≥ 1

Savings for Witten-Bell

- No need to store singleton suffix counts
- Storage $\approx 1.3\times$ storage for standard counts



Applying Related Bounds

Actual Error Rate

Errors only occur if we query for nonexistent counts

$$Pr(\text{Actual Error}) = Pr(x \notin \text{Corpus} | x \in \text{Hypothesis}) \times f$$

Can we increase the *a priori* membership probability?

Monotonicity of N -gram Event Space

- If a unigram x tests false, then a bigram xy cannot be a member
- In general,

$$freq(xy) \leq \min\{freq(x), freq(y)\}$$



An Example

Interpolated Witten-Bell BF-LM

$$P_{wb}(w_i | w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} P_{ml}(w_i | w_{i-n+1}^{i-1}) \\ + (1 - \lambda_{w_{i-n+1}^{i-1}}) P_{wb}(w_i | w_{i-n+2}^{i-1})$$

where λ_x is defined via,

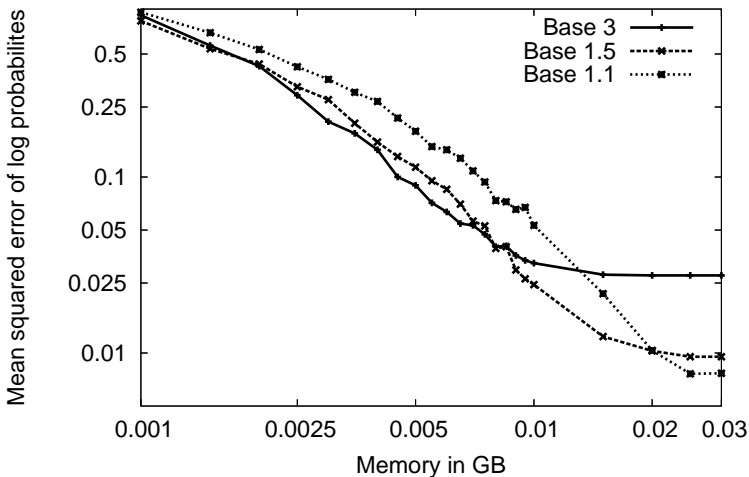
$$1 - \lambda_x = \frac{\text{count}(x)}{\text{suffix}(x) + \text{count}(x)},$$

- Start from lowest order event (i.e. unigram)
- Bound numerator in ml term by count of denominator
- Bound suffix count by its token frequency
- Truncate computation if ml denominator is zero



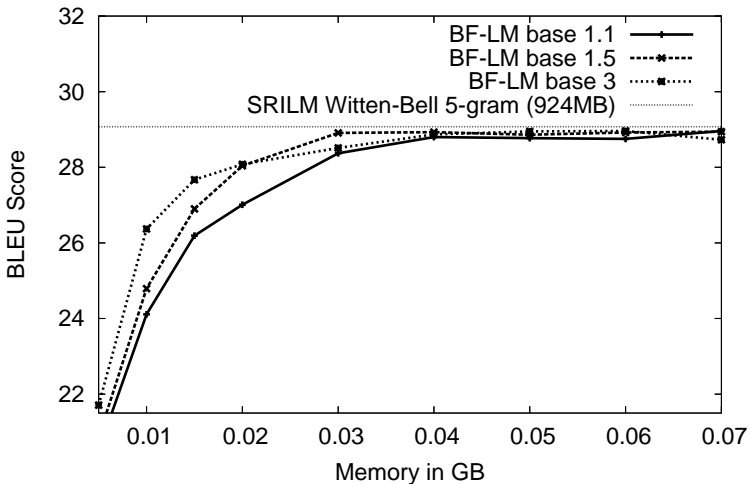
Mean Squared Error between Lossless and BFLM

MSE between WB 3-gram SRILM and BF-LMs



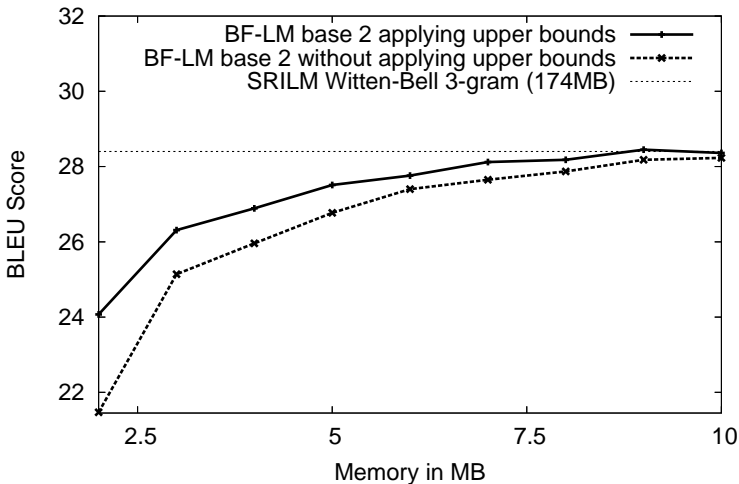
BLEU Score with WB 5-gram BFLM

WB-smoothed BF-LM 5-gram model



Effect of Applying Upper Bounds

WB-smoothed BF-LM 3-gram model



Summary

- Log-Frequency BF can be used to store approximate corpus statistics **very efficiently**
- Smoothed LMs derived from log-freq BFs can approximate lossless LMs with around **15 bits** per N -gram
- Future Work
 - More efficient BF counting schemes
 - Hybrid models - e.g. explicit 1,2-grams + BF 3,4,5-grams
 - Other NLP applications of log-frequency BF framework



Some References



R. Motwani and P. Raghavan.

Randomized Algorithms.

Cambridge University Press, 1995.



B. Bloom.

Space/time tradeoffs in hash coding with allowable errors.

Communications of the ACM, 13:422–426, 1970.



A. Broder and M. Mitzenmacher.

Network Applications of Bloom filters: A Survey.

Internet Mathematics, 1(4):485–509, 2005.



G. Cormode and S. Muthukrishnan.

An improved data stream summary: the CountMin sketch and its applications.

Journal of Algorithms, 55(1):58–75, April 2005.



Thanks

- Thanks for listening!
- Thanks to all the Moses Team!

