

Data structures in Information Retrieval

Language Models and Graphs

Speller example

Britany Spears

Britany → Britain (LD: 2 Mod. Pr.: $5e-5$)

Britany → Britney (LD: 2 Mod. Pr.: $3e-7$)

Britain Spears (Pr: $7e-8$)

Britney Spears (Pr: $3e-4$)

Language models

Language model – probabilistic presentation of language:

- Characters
- Words
- Phrases

Language Models Applications

- Index/Search pruning
- Distribution Data in cluster/ balancing
- Spelling/ Query Suggestion
- Ranking Algorithm
- Classification/extraction/ other IR tasks

N-gram word model

$(\text{Word}_1, \text{Word}_2, \dots, \text{Word}_n) \rightarrow P$

Smoothing:

- Unseen N-grams

Laplace: $P = 1/(n+2)$

- Outliers, lack of data

Prior: Poisson, Dirichlet

Naïve Presentation of LM

Key	Prob
Word1	0.002
Word2	0.0004
WordN	0.00043

Key	Prob
Word1, Word'1	0.000001
Word2, Word'2	0.000001
WordN, Word'N	0.00003

Any search structure



Building LM

Влияние морфологического анализа на качество информативного поиска

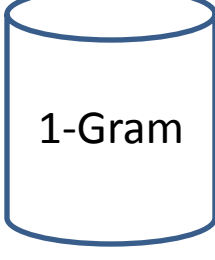
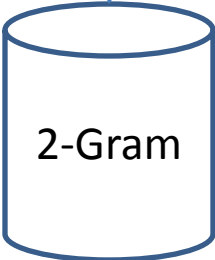
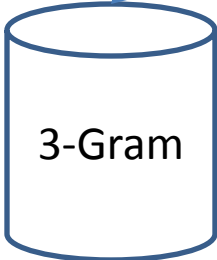
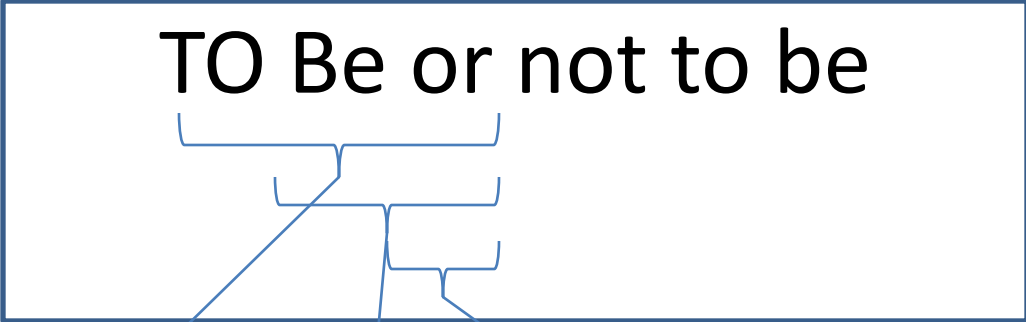
© М.В. Губин А.В. Морозов
 Екатеринбург: Клоузет
www.klozet.ru or info@klozet.ru

Аннотация
 Целью работы является исследование влияния морфологического анализа на качество информативного поиска. В работе рассмотрены различные подходы к обработке форм, включая как традиционные методы, так и современные методы, основанные на использовании статистических моделей. Проведено сравнение результатов поиска с использованием морфологического анализа и без него. Показано, что использование морфологического анализа повышает качество поиска, особенно в случаях, когда в запросе используются различные формы слов. Для оценки качества поиска использованы различные метрики, включая точность, полноту и F-меру. Результаты работы показывают, что использование морфологического анализа повышает качество поиска на 10-15%.

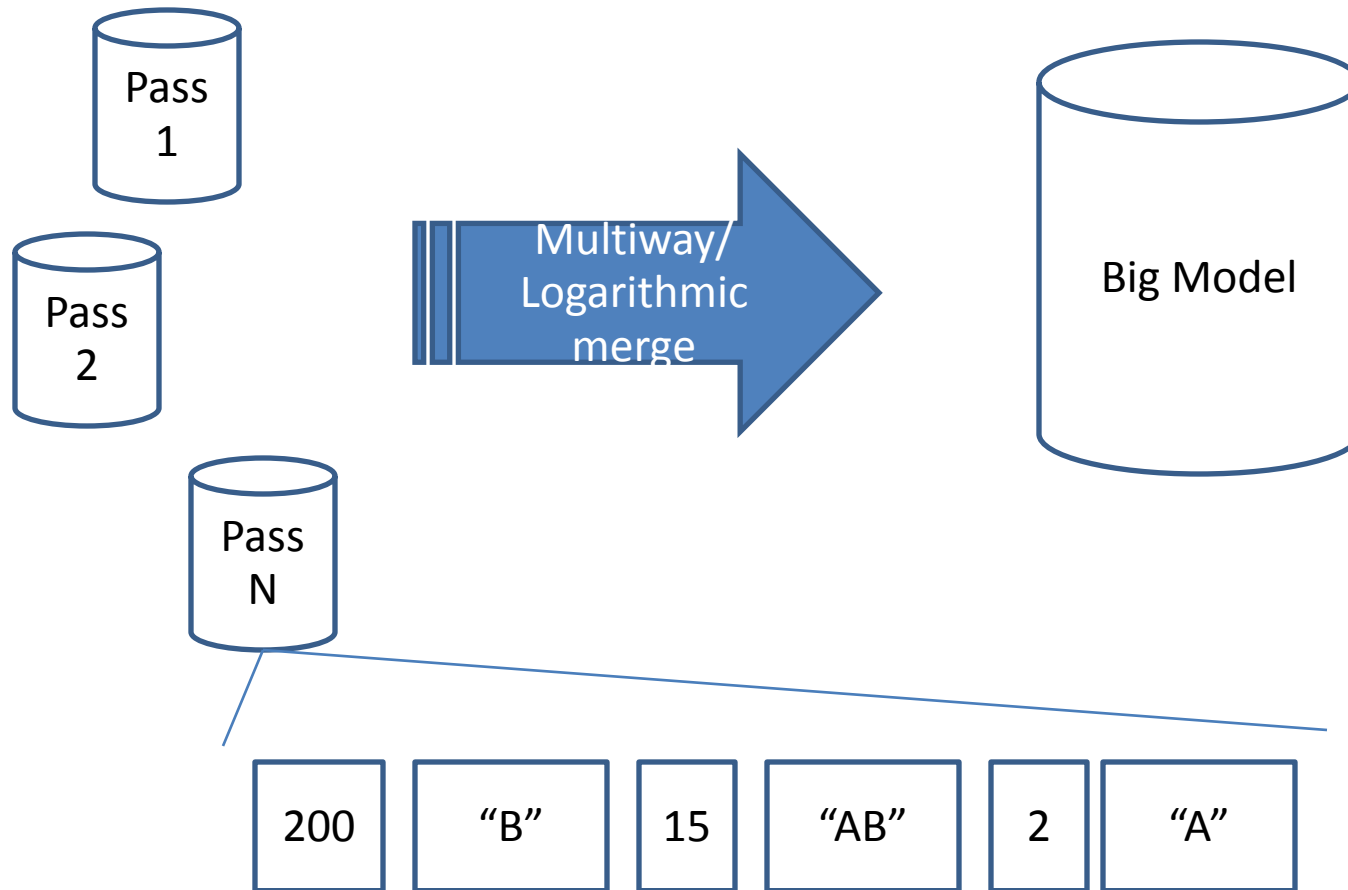
Введение
 В настоящее время активно ведутся исследования в области информативного поиска. Одним из ключевых направлений является исследование влияния морфологического анализа на качество поиска. В данной работе рассмотрены различные подходы к обработке форм, включая как традиционные методы, так и современные методы, основанные на использовании статистических моделей. Проведено сравнение результатов поиска с использованием морфологического анализа и без него. Показано, что использование морфологического анализа повышает качество поиска, особенно в случаях, когда в запросе используются различные формы слов. Для оценки качества поиска использованы различные метрики, включая точность, полноту и F-меру. Результаты работы показывают, что использование морфологического анализа повышает качество поиска на 10-15%.

Использование морфологического анализа в поисковой системе
 В поисковой системе важно учитывать морфологические особенности слов. Это позволяет находить документы, содержащие различные формы одного и того же слова. Для этого используются различные методы, включая как традиционные методы, так и современные методы, основанные на использовании статистических моделей. Проведено сравнение результатов поиска с использованием морфологического анализа и без него. Показано, что использование морфологического анализа повышает качество поиска, особенно в случаях, когда в запросе используются различные формы слов. Для оценки качества поиска использованы различные метрики, включая точность, полноту и F-меру. Результаты работы показывают, что использование морфологического анализа повышает качество поиска на 10-15%.

Linearization



Big LM



Presentation of Probabilities

During build: counters

During use: $\log(p)$

$$P_1 * P_2 \dots * P_N \rightarrow \log(P_1) + \log(P_2) \dots + \log(P_N)$$

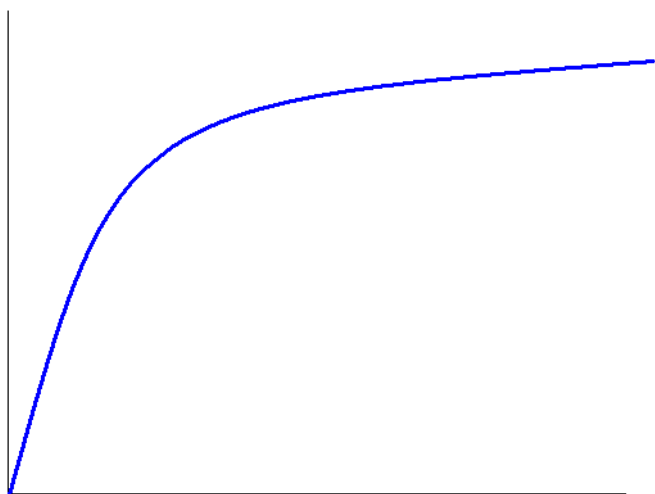
$$P_1 + P_2 \dots + P_N \rightarrow \log(e^{\log(P_1)} + e^{\log(P_2)} \dots + e^{\log(P_N)}) =$$

$$\log(P_1) + \log(1 + e^{\log(P_2) - \log(P_1)} \dots + e^{\log(P_N) - \log(P_1)})$$

Example of “super-pruned LM”

“the”=0.062 “of” = 0.03 others = 0.000028
H=0.28

“the” = 0.062 others = 0.000029 H=0.17



Pruning LMs

Task: Put into memory N-gram model

Model: N words + probability

1. Create N-1, N-2... word model
2. Use low order model to predict high order [backoffs]
3. Pruning: $|P_{\text{est}} - P_{\text{real}}| < C \Rightarrow$ remove [iterative]
4. Perform lossy compression


Use low order model for prediction

$$P(W1) = X$$

$$P(W2) = Y$$

$$P(W1, W2) = P(W1) * P(W2 | W1) \approx X * Y$$

Independent assumption
 $P(W2|W1) \approx P(W2)$



Lossy compression of LM (1)

1. Take a good hash function

$F(w_1, \dots, w_N) \rightarrow [0 \dots \text{BIG_INT})$

BIG_INT reduces noise

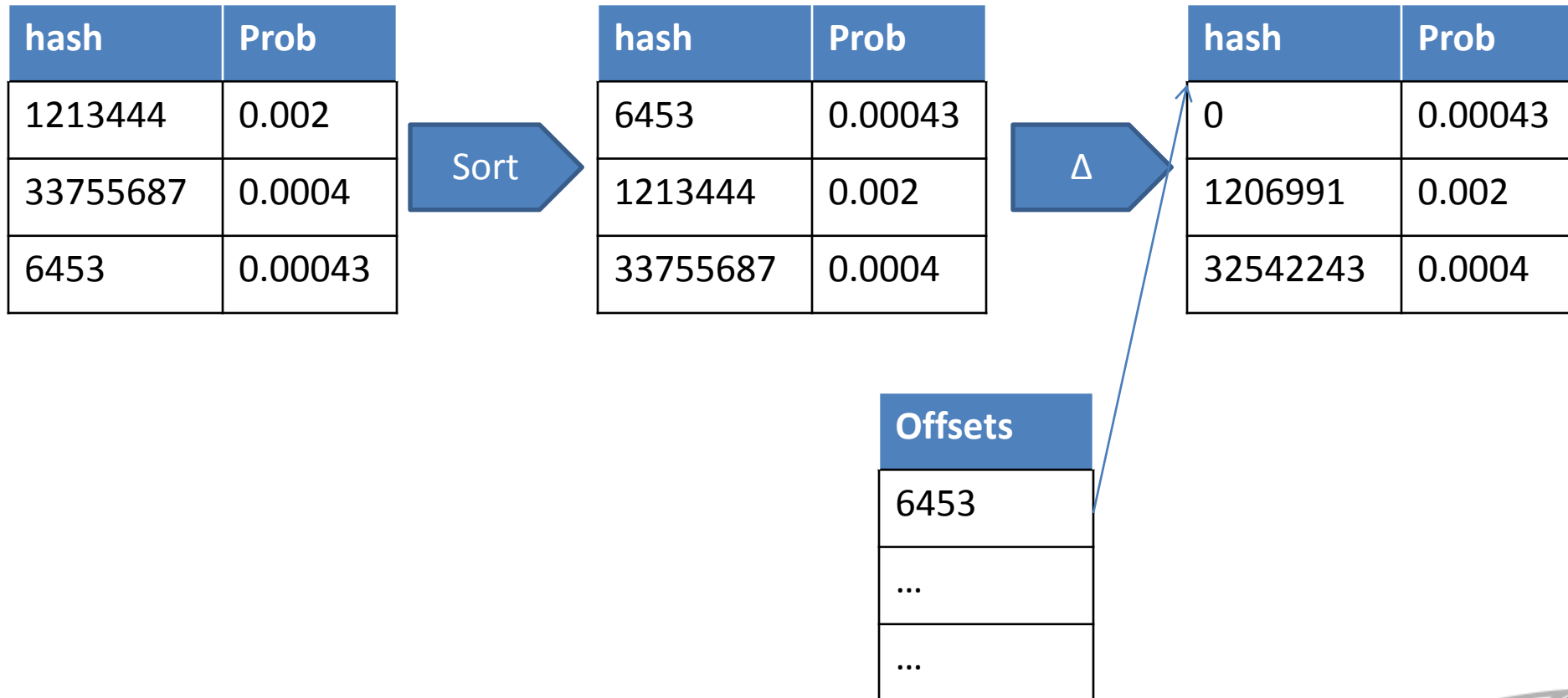
2. Encode Model:

$\langle \text{to, be: } 0.002 \rangle \rightarrow \langle 1213444: 0.002 \rangle$

$\langle \text{be, or: } 0.0004 \rangle \rightarrow \langle 33755687: 0.0004 \rangle$

$\langle \text{or, not: } 0.00043 \rangle \rightarrow \langle 6453: 0.00043 \rangle$

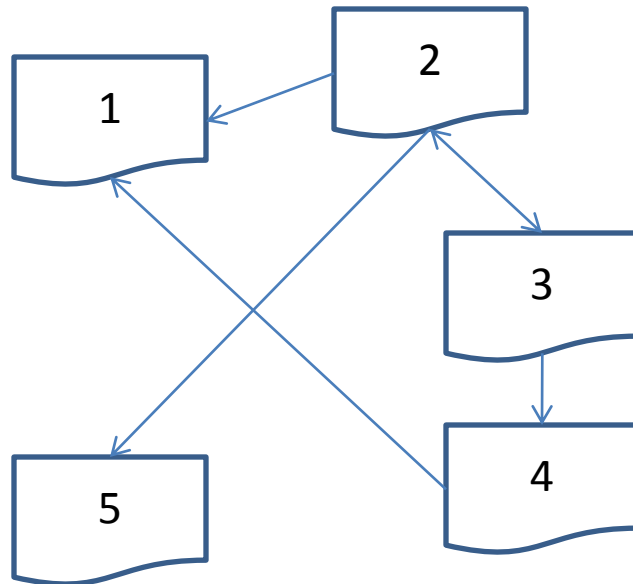
Lossy compression of LMs (2)



Using compressed LM

```
hash = hash_function(phrase)
(offset,n_offset) = search_in_offsets(hash)
for decom in decompression(offset,n_offset):
    if hash = decom.cod:
        return decom.prob
backup(phrase)
```

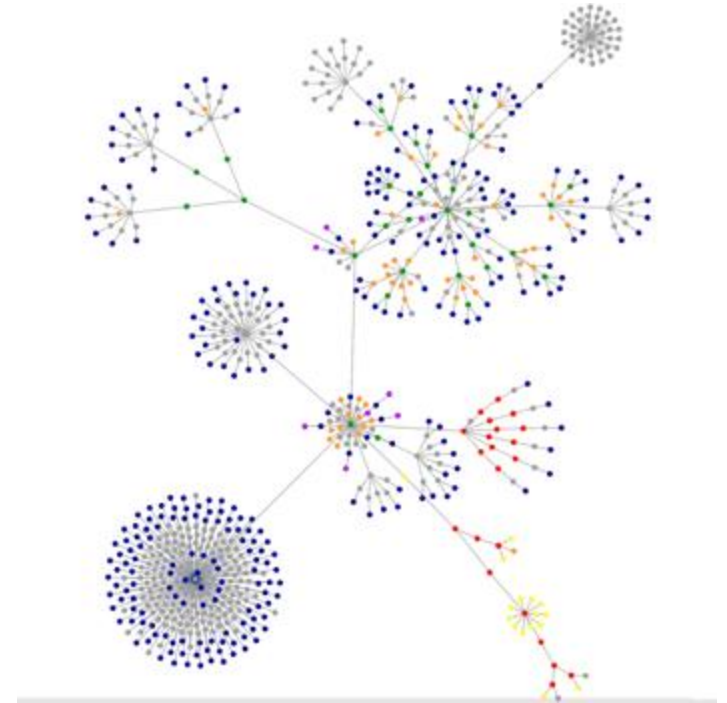
Links Analysis



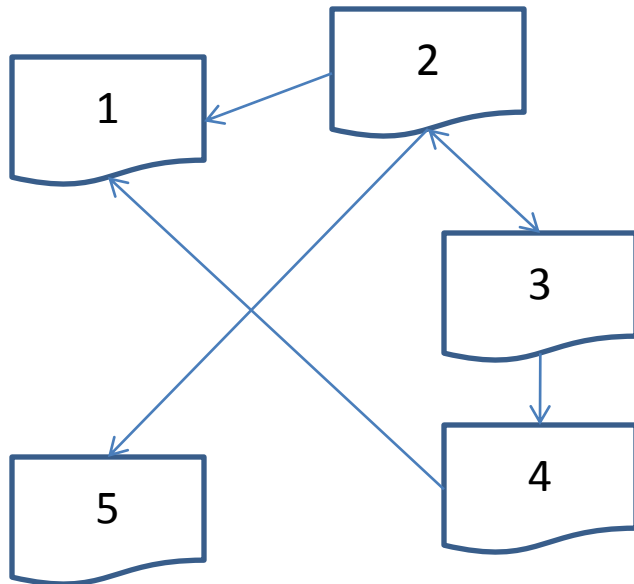
ID	1	2	3	4	5
1					
2	1		1		1
3		1		1	
4	1				
5					

Sparseness of Adjacency Matrix

- Small-world phenomena
- Power-law distribution



Adjacency Matrix Presentation



1 → <>

2 → <1,3,5>

3 → <2,4>

4 → <1>

5 → <>

Compress with all algorithms from Inverted Files!

Webgraph <http://law.dsi.unimi.it/> ζ codes 3.08 bits/link

PageRank Algorithm

“Probability of random walk”

“Popularity metrics”

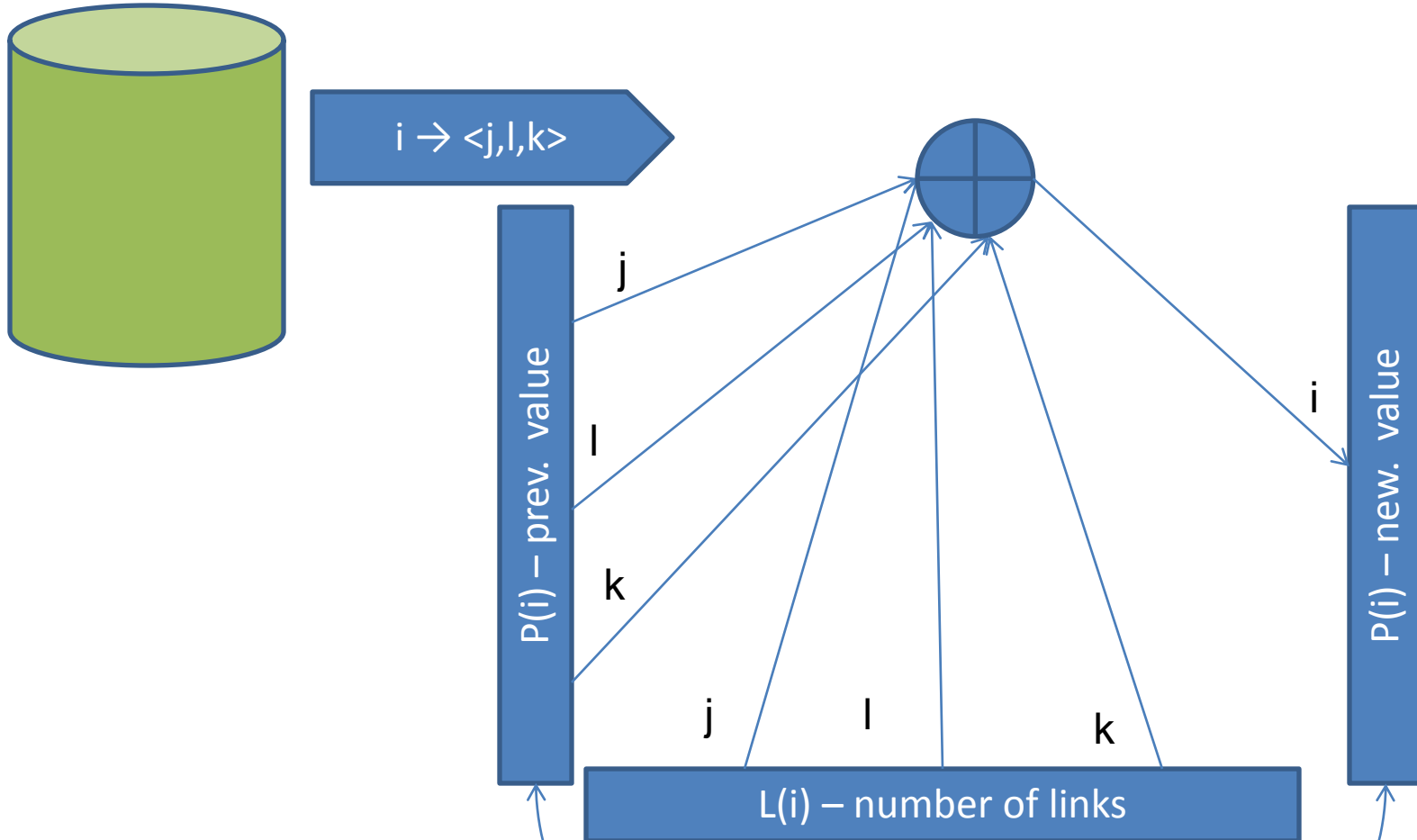
Eigenvector $Ax = \lambda x$

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

Diagram illustrating the PageRank formula with annotations:

- Initial value**: Points to the term $\frac{1-d}{N}$.
- Dumping factor**: Points to the variable d .
- Number of links**: Points to the denominator $L(p_j)$ in the summation term.

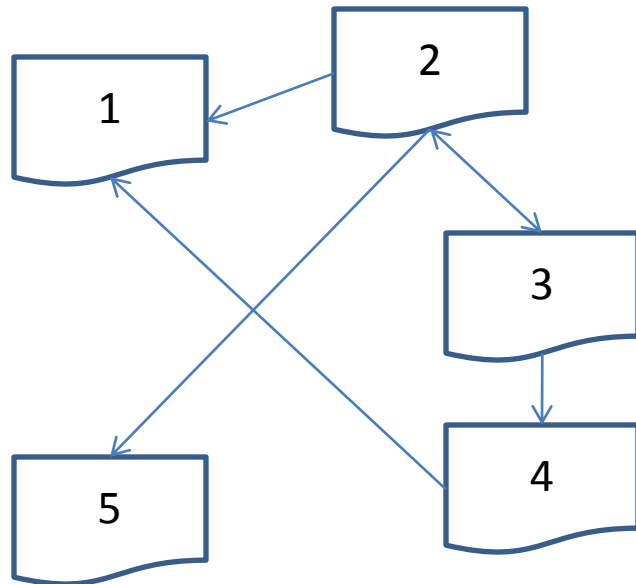
Pager Rank Implementation



$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

PageRank

optimization 1 (1)



Inverted Adjacency Matrix
Presentation

1 \rightarrow <2,4>

2 \rightarrow <3>

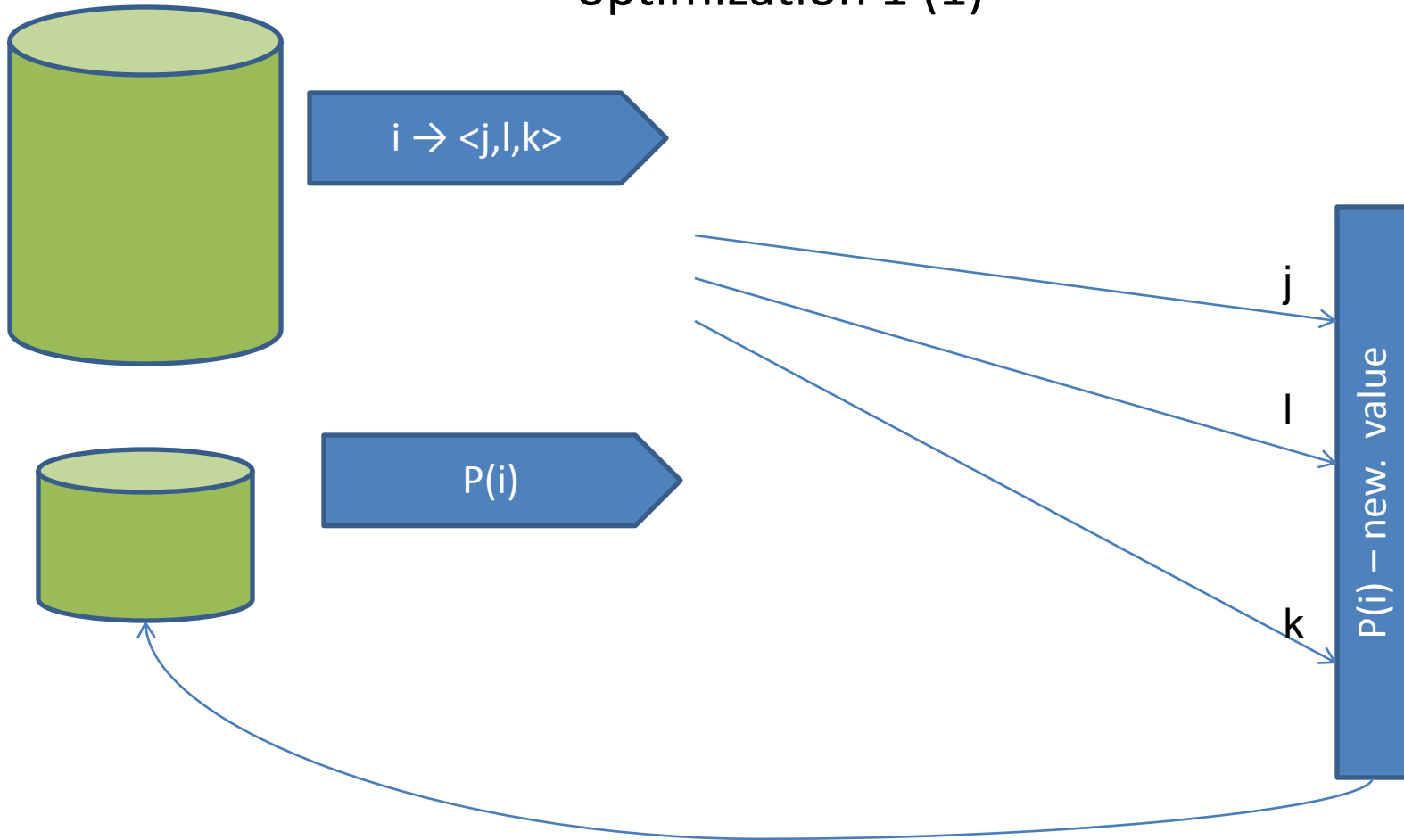
3 \rightarrow <2>

4 \rightarrow <3>

5 \rightarrow <2>

PageRank

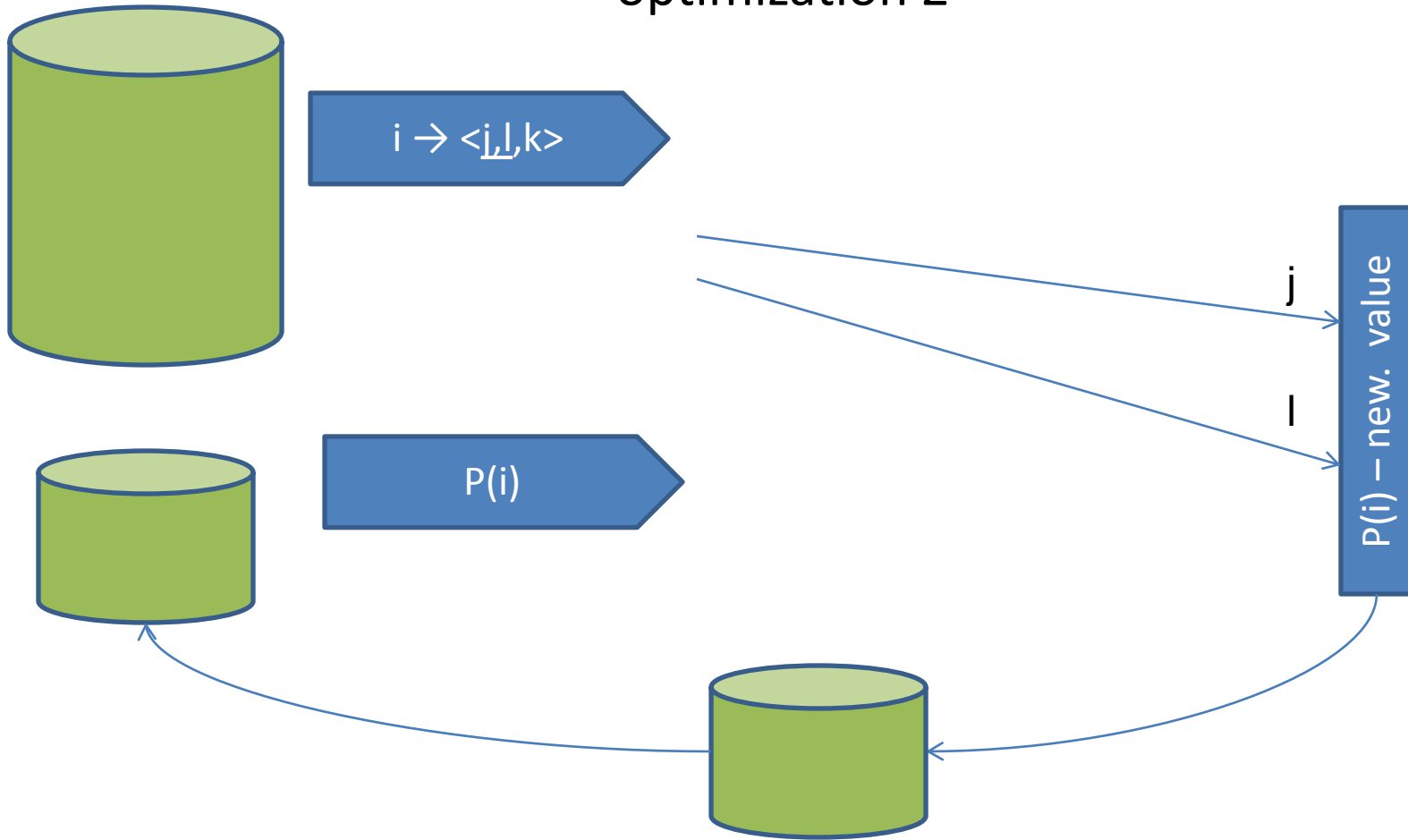
optimization 1 (1)



$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

PageRank

optimization 2

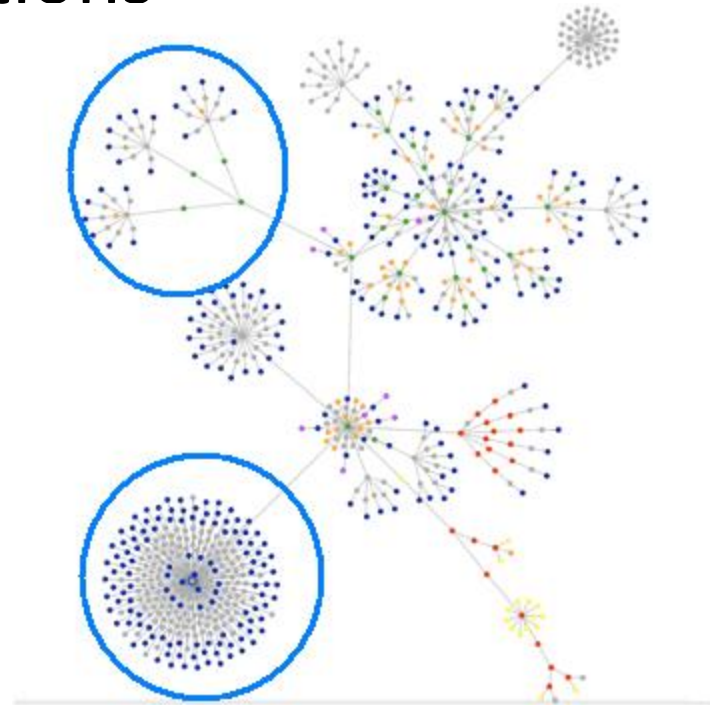


Reduce size of array

PageRank

optimization ...

1. Reduce number of iterations: initial values from previous calculations
2. Parallelization
3. Graph reduce



Converge conditions

1. Fixed number of iterations (<100)
2. $\sum (P_{\text{new}}(i) - P(i)) < \Delta$
3. No re-ranking:
 $P(i) > P(j) \rightarrow P_{\text{new}}(i) > P_{\text{new}}(j)$

Variants: N-top, sample

Finished

1. Index big corpus
2. Search in index
3. LM for spell-corrector
4. Link analysis

Q&A

References

1. **Emmanuel Eckard and Jean-C´edric Chappelier** Free Software for research in Information Retrieval and Textual Clustering **Technical report 2007**
2. **Ricardo Baeza-Yates et al.** Modern Information Retrieval ACM Press New York 1999
3. **Manning et al.** An Introduction to Information Retrieval Cambridge University Press (July 31, 2008)
4. **Donald E. Knuth** The Art of Computer Programming Vol. 3 Sorting and Searching , Second Edition Addison-Wesley, 1998
5. **Ulrich Drepper** What Every Programmer Should Know About Memory Red Hat, Inc 2007 <http://people.redhat.com/drepper/cpumemory.pdf>
6. **Witten et al.**, Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd edition published by Morgan Kaufmann Publishing, San Francisco, ISBN 1-55860-570-3

References

7. **Cheng, C., Shann, J. J., and Chung, C.** Unique-order interpolative coding for fast querying and space-efficient indexing in information retrieval systems. *Inf. Process. Manage.* 42, 2 (Mar. 2006), 407-428.
8. **Sergey Melnik, Sriram Raghavan, Beverly Yang, Hector Garcia-Molina** Building a Distributed Full-Text Index for the Web Proceedings *WWW10, 2001*
9. **S. Buttcher, C. Clarke** A Document-Centric Approach to Static Index Pruning in Text Retrieval Systems, *CIKM'06*
10. **Diego Puppini, Fabrizio Silvestri, Domenico Laforenza.** "Query-Driven Document Partitioning and Collection Selection". Invited Paper. Proceedings of *INFOSCALE '06*
11. **Ken Church et al.** Compressing Trigram Language Models With Golomb Coding Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational

References

12. **Gianna M. Del Corso, Antonio Gullí, and Francesco Romani** Fast PageRank Computation via a Sparse Linear System Internet Mathematics Vol. 2, No. 3: 251-273
13. **Frank McSherry** A Uniform Approach to Accelerated PageRank Computation Proceedings WWW 2005 conference
14. **Taher Haveliwala**. "Efficient Computation of PageRank," Stanford University Technical Report, September 1999