

# A Time Machine for Text Search

Klaus Berberich, Srikanta Bedathur,  
Thomas Neumann, Gerhard Weikum

Max-Planck Institute for Informatics,  
Saarbrücken, Germany



# Motivation

- Historical information needs, e.g.,

Contemporary (~2001) articles about the movie “*Harry Potter and the Sorcerer’s Stone*”

- Relevant pages have **disappeared** but are **preserved by Web archives** (e.g., archive.org)
- Search over Web archives is **limited** and **ignores the time-axis**



# Motivation

The screenshot shows a search engine interface with the following elements:

- Search Bar:** Contains the text "harry potter and the sorcerer's".
- Search Results:** A list of links with titles and snippets. The first result is "Harry Potter and the Sorcerer's Stone (2001)" from IMDb, with a snippet: "IMDb > Harry Potter and the Sorcerer's Stone (2001) ... Harry Potter and the Philosopher's Stone (Canada: English title) (International: ... us.imdb.com/Title?0241527 · Cached".
- Left Sidebar:** Contains navigation links like "Web | Images | City | News | More", a search box with the same text, and sections for "Narrow Your Search" (listing various Harry Potter titles and topics) and "Expand Your Search" (listing "Hogwarts", "Quidditch", etc.).
- Bottom:** A "Related Names" section listing "Emma Watson" and "Daniel Radcliffe".



# Motivation

- Historical information needs, e.g.,

Contemporary (~2001) articles about the movie “*Harry Potter and the Sorcerer’s Stone*”

- Relevant pages have **disappeared** but are **preserved by Web archives** (e.g., archive.org)
- Search over Web archives is **limited** and **ignores the time-axis**



# Motivation

http://web.archive.org/web/20020209030855/www.nytimes.com/2001/11/16/movies/16POTT.html

Wizard School Without the ...

**Movies** The New York Times

HOME | Search | Past 30 Days | Welcome, [ia\\_archiver](#)

[Go to Advanced Search](#) | [Sign Up for Newsletters](#) | [Log Out](#)

[E-Mail This Article](#) | [Printer-Friendly Format](#)

[Most E-Mailed Articles](#) | [Single-Page View](#)

NEWS

International  
National  
Nation Challenged  
Politics  
Business  
Technology  
Science  
Health  
Sports  
New York Region  
Education  
Weather  
Obituaries  
NYT Front Page  
Connections  
Special Winter  
Olympics

OPINION

Editorials/Op-Ed  
Readers' Opinions

**Scotland, PA. Opens Today**

FEATURES

Arts  
Books  
Movies  
Travel  
Dining & Wine  
Home & Garden  
Fashion & Style  
New York Today  
Crossword/Games  
Cartoons  
Magazine  
Week in Review  
Photos  
College  
Learning Network

SERVICES

Archive  
Classifieds  
Help Center  
NYT Mobile  
NYT Store  
E-Cards & More  
About NYTDigital  
Jobs at NYTDigital  
Online Media Kit  
Our Advertisers

NEWSPAPER

Home Delivery  
Customer Service  
Electronic Edition  
Media Kit

November 16, 2001

MOVIE REVIEW | 'HARRY POTTER AND THE SORCERER'S STONE'

**Wizard School Without the Magic**

By ELVIS MITCHELL

**T**HE world may not be ready yet for the film equivalent of books on tape, but this peculiar phenomenon has arrived in the form of the film adaptation of J. K. Rowling's "Harry Potter and the Sorcerer's Stone." The most highly awaited movie of the year has a dreary, literal-minded competence, following the letter of the law as laid down by the author. But it's all muted flourish, with momentary pleasures, like Gringott's, the bank staffed by trolls that looks like a Gaudí throwaway. The picture is so careful that even the tape wrapped around the bridge of Harry's glasses seems to have come out of the set design. (It never occurred to anyone to show him taping the frame together.)

The movie comes across as a covers act by an extremely competent tribute band not the real thing but an incredible simulation and there's an audience for this sort of thing. But watching "Harry Potter" is like seeing "Beatlemania" staged in the Hollywood Bowl, where the cheers and screams

ADVERTISEMENT

Check Delivery Options | 50% Off-Click Here!

les about the  
orcerer's Stone”

but are preserved

ed and ignores the



# Motivation

- Historical information needs, e.g.,

Contemporary (~2001) articles about the movie “*Harry Potter and the Sorcerer’s Stone*”

- Relevant pages have **disappeared** but are **preserved by Web archives** (e.g., archive.org)
- Search over Web archives is **limited** and **ignores the time-axis**



# Motivation

- **Time-Travel Text Search** extends keyword querying by a time-point of interest  $t$

*“harry potter” @ 2001/11/14*

- Other **temporally versioned text collections**
  - Wikis
  - Repositories (e.g., controlled by CVS, Subversion)
  - Your Desktop



# Outline

- Motivation
- **Collection, Query, and Relevance Model**
- **Time-Travel Inverted File Index**
  - Reducing Index Size
  - Tuning Index Performance
- Experimental Evaluation
- Conclusions



# Collection Model

- Document  $d$  is a sequence of time-stamped versions

$$\mathbf{d} = \langle d^{t_1}, d^{t_2}, \dots \rangle$$

- Version is a vector of searchable terms
- Document deletion results in tombstone version  $\perp$
- Discrete time, timestamps are non-negative
- State of document collection as of time  $t$

$$\mathbf{D}^t = \bigcup_{\mathbf{d} \in \mathbf{D}} \{d^{t_i} \in \mathbf{d} \mid t \in \text{val}(d^{t_i}) \wedge d^{t_i} \neq \perp\}$$

# Query Model

- Time-travel query  $q^t$  consists of
  - keyword part  $q$  (i.e., a set of query terms)
  - time-point of interest  $t$
- Time-travel query  $q^t$  is evaluated over  $D^t$  so that only versions that existed at time  $t$  are considered



# Relevance Model

- We adapt **Okapi BM25** as a relevance model

$$w(q^t, d^{t_i}) = \sum_{v \in q} w_{tf}(v, d^{t_i}) \cdot w_{idf}(v, t)$$

- **Term-frequency** score (TF)

$$w_{tf}(v, d^{t_i}) = \frac{(k_1 + 1) \cdot tf(v, d^{t_i})}{k_1 \cdot ((1 - b) + b \cdot \frac{dl(d^{t_i})}{avdl(t_i)}) + tf(v, d^{t_i})}$$

- **Inverse document-frequency** score (IDF)

$$w_{idf}(v, t) = \log \frac{N(t) - df(v, t) + 0.5}{df(v, t) + 0.5}$$

# Relevance Model

- We adapt **Okapi BM25** as a relevance model

$$w(q^t, d^{t_i}) = \sum_{v \in q} w_{tf}(v, d^{t_i}) \cdot w_{idf}(v, t)$$

- **Term-frequency score (TF)**

$$w_{tf}(v, d^{t_i}) = \frac{(k_1 + 1) \cdot tf(v, d^{t_i})}{k_1 \cdot ((1 - b) + b \cdot \frac{dl(d^{t_i})}{avdl(t_i)}) + tf(v, d^{t_i})}$$

- **Inverse document-frequency score (IDF)**

$$w_{idf}(v, t) = \log \frac{N(t) - df(v, t) + 0.5}{df(v, t) + 0.5}$$

# Relevance Model

- We adapt **Okapi BM25** as a relevance model

$$w(q^t, d^{t_i}) = \sum_{v \in q} w_{tf}(v, d^{t_i}) \cdot w_{idf}(v, t)$$

- **Term-frequency** score (TF)

$$w_{tf}(v, d^{t_i}) = \frac{(k_1 + 1) \cdot tf(v, d^{t_i})}{k_1 \cdot ((1 - b) + b \cdot \frac{dl(d^{t_i})}{avdl(t_i)}) + tf(v, d^{t_i})}$$

- **Inverse document-frequency** score (IDF)

$$w_{idf}(v, t) = \log \frac{N(t) - df(v, t) + 0.5}{df(v, t) + 0.5}$$

# Relevance Model

- We adapt **Okapi BM25** as a relevance model

$$w(q^t, d^{t_i}) = \sum_{v \in q} w_{tf}(v, d^{t_i}) \cdot w_{idf}(v, t)$$

- **Term-frequency** score (TF)

$$w_{tf}(v, d^{t_i}) = \frac{(k_1 + 1) \cdot tf(v, d^{t_i})}{k_1 \cdot ((1 - b) + b \cdot \frac{dl(d^{t_i})}{avdl(t_i)}) + tf(v, d^{t_i})}$$

- **Inverse document-frequency** score (IDF)

$$w_{idf}(v, t) = \log \frac{N(t) - df(v, t) + 0.5}{df(v, t) + 0.5}$$

# Relevance Model

- We adapt **Okapi BM25** as a relevance model

$$w(q^t, d^{t_i}) = \sum_{v \in q} w_{tf}(v, d^{t_i}) \cdot w_{idf}(v, t)$$

- **Term-frequency** score (TF)

$$w_{tf}(v, d^{t_i}) = \frac{(k_1 + 1) \cdot tf(v, d^{t_i})}{k_1 \cdot ((1 - b) + b \cdot \frac{dl(d^{t_i})}{avdl(t_i)}) + tf(v, d^{t_i})}$$

- **Inverse document-frequency** score (IDF)

$$w_{idf}(v, t) = \log \frac{N(t) - df(v, t) + 0.5}{df(v, t) + 0.5}$$

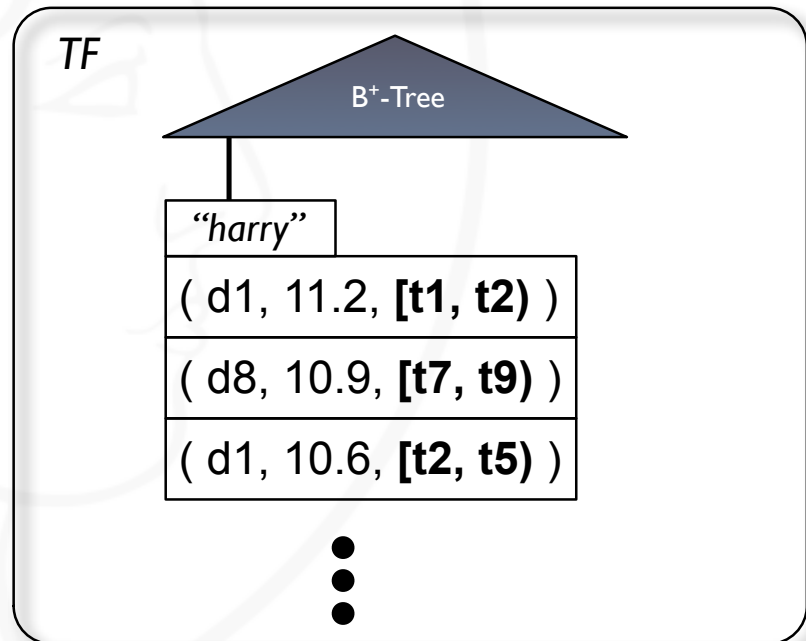
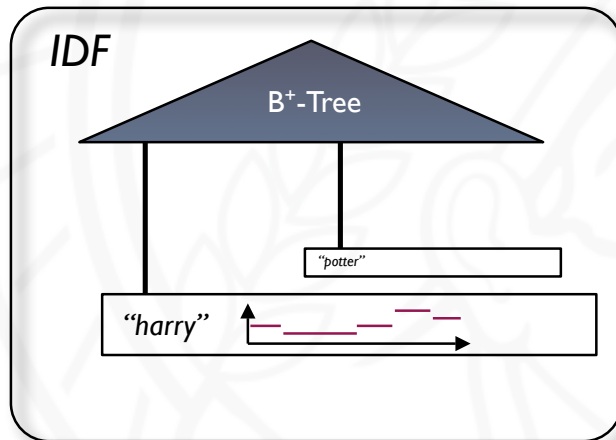
# Outline

- Motivation
- Collection, Query, and Relevance Model
- **Time-Travel Inverted File Index**
  - Reducing Index Size
  - Tuning Index Performance
- Experimental Evaluation
- Conclusions



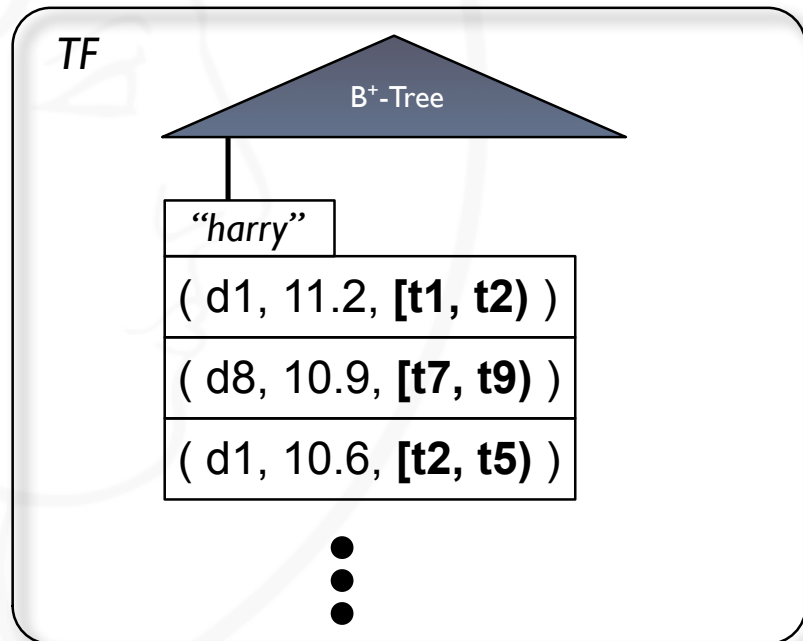
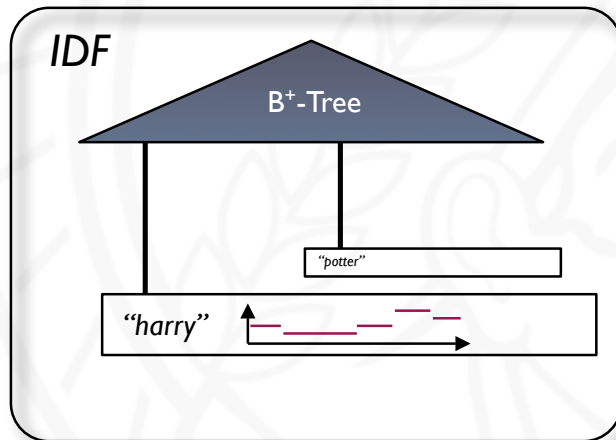
# Time-Travel Inverted File Index

- Idea: Transparently extend “IR’s **workhorse**” so that the existing wealth of extensions remains applicable
- We extend postings by a **validity time-interval**



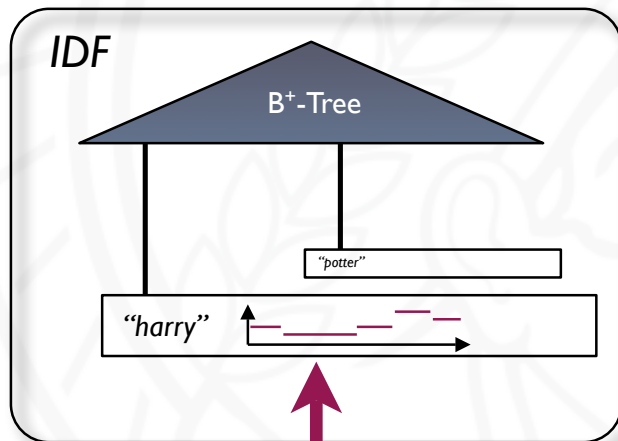
# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

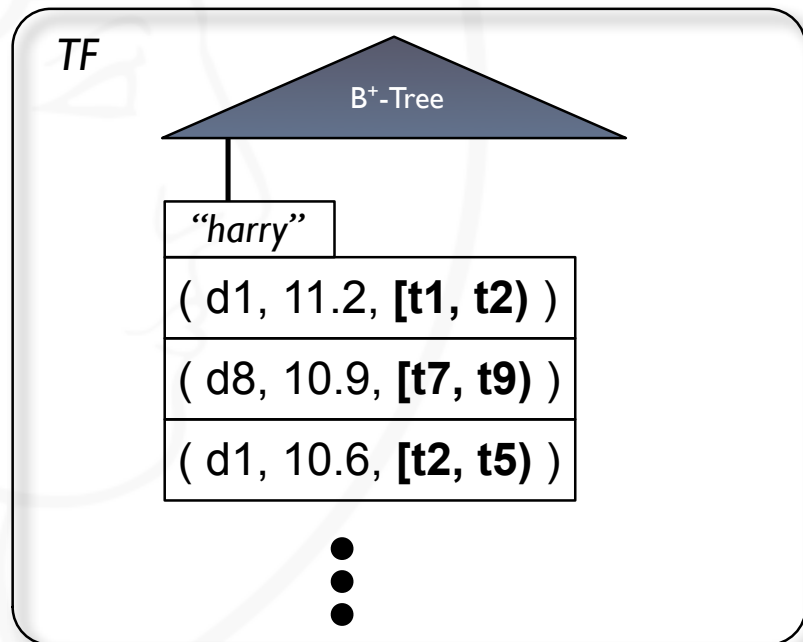


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

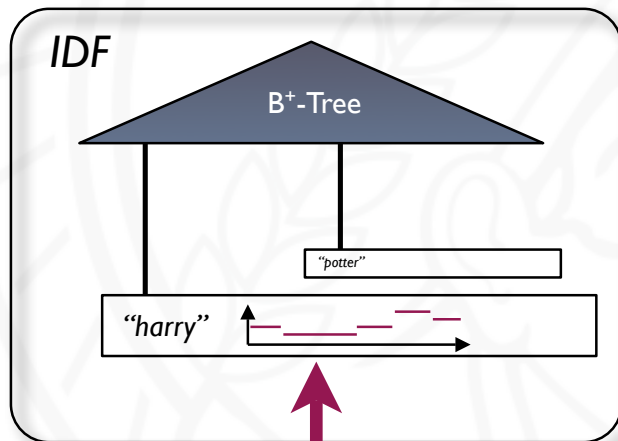


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

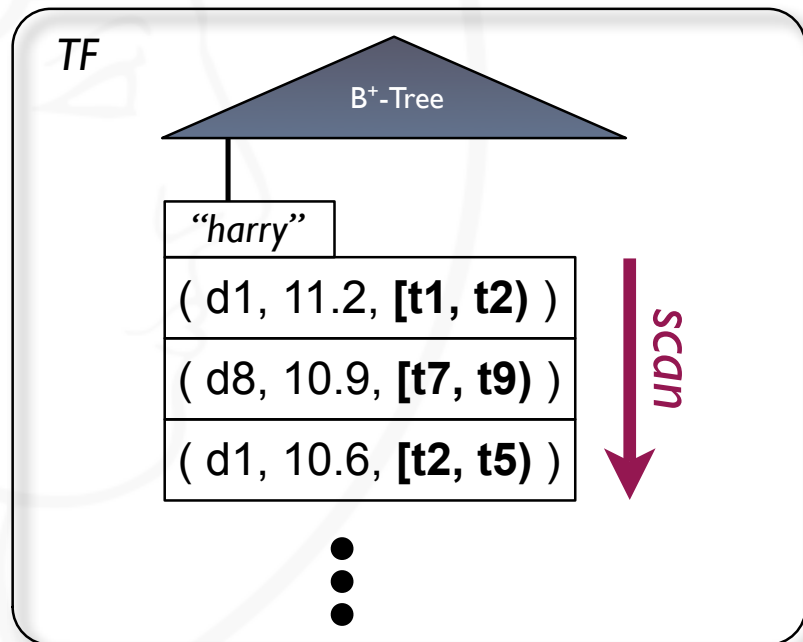


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

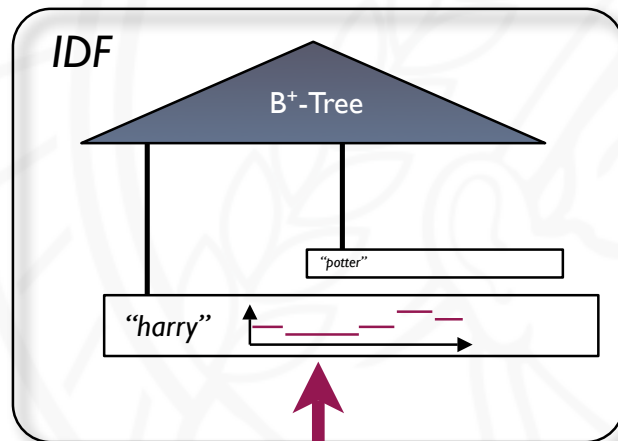


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

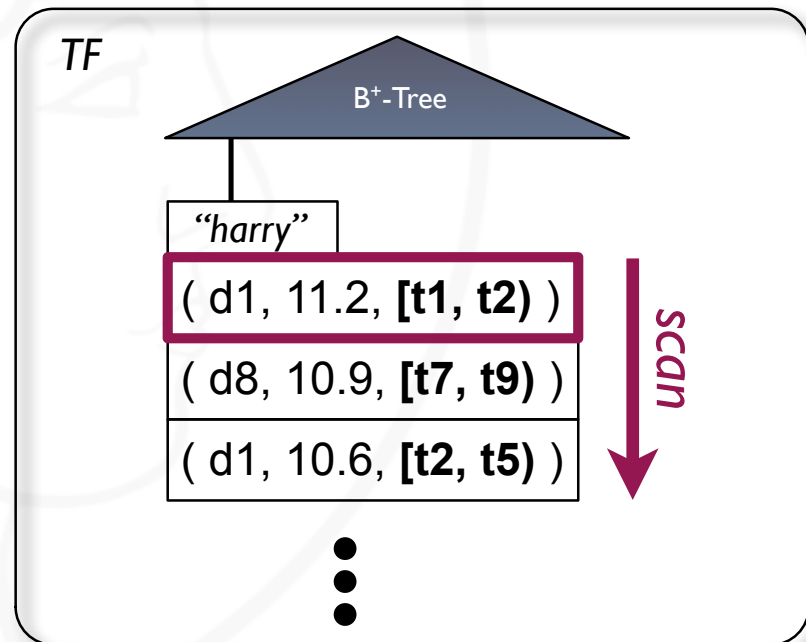


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

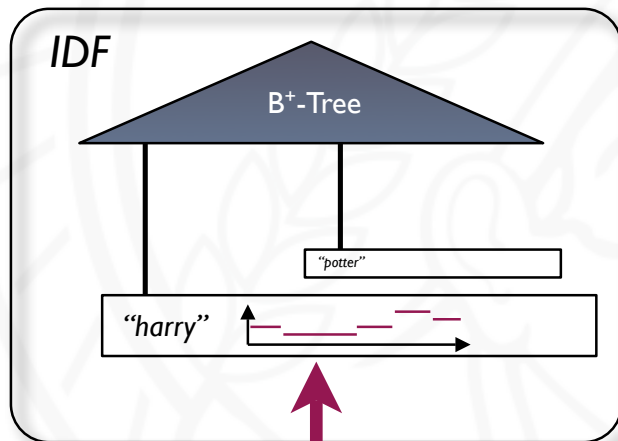


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

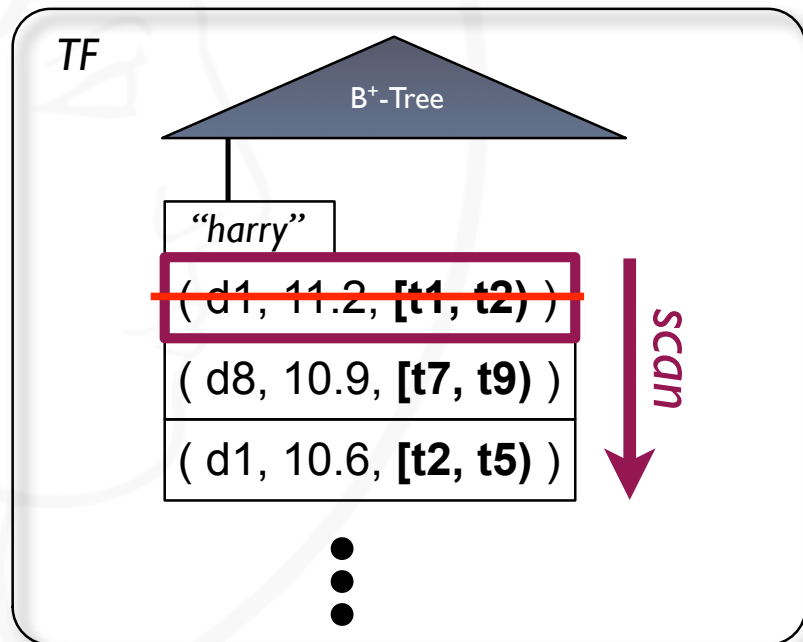


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

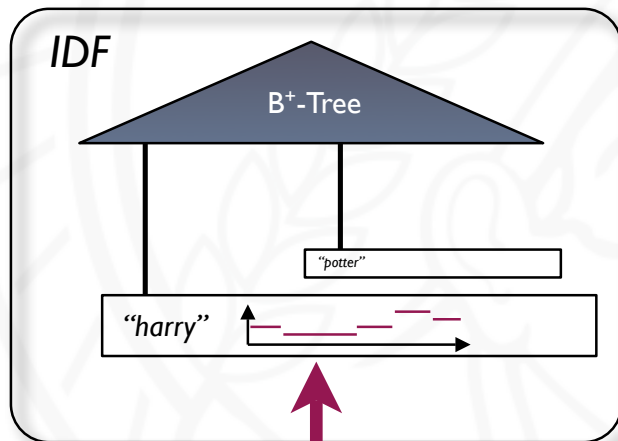


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

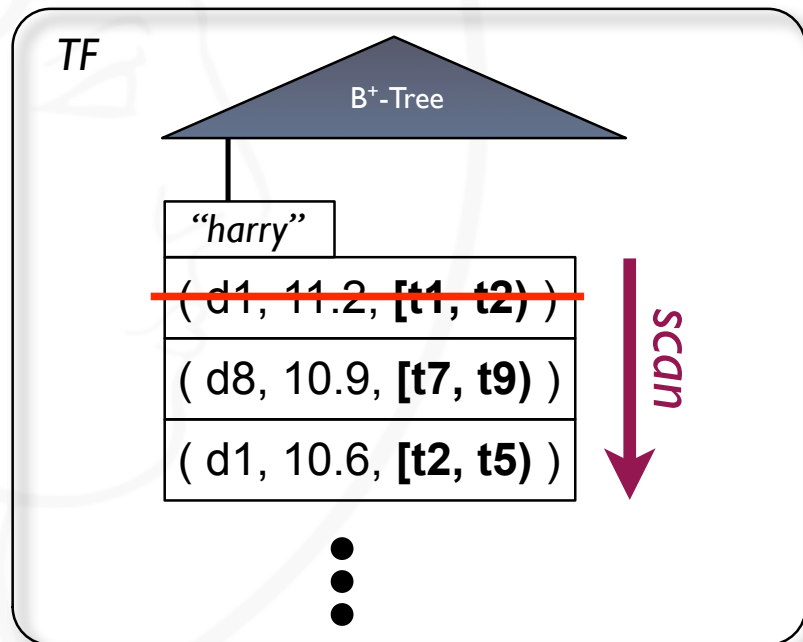


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

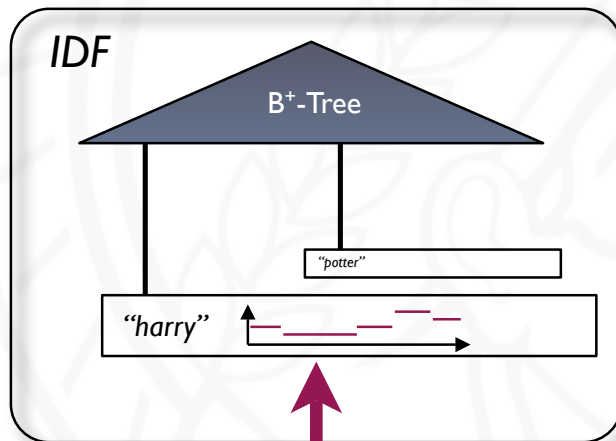


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

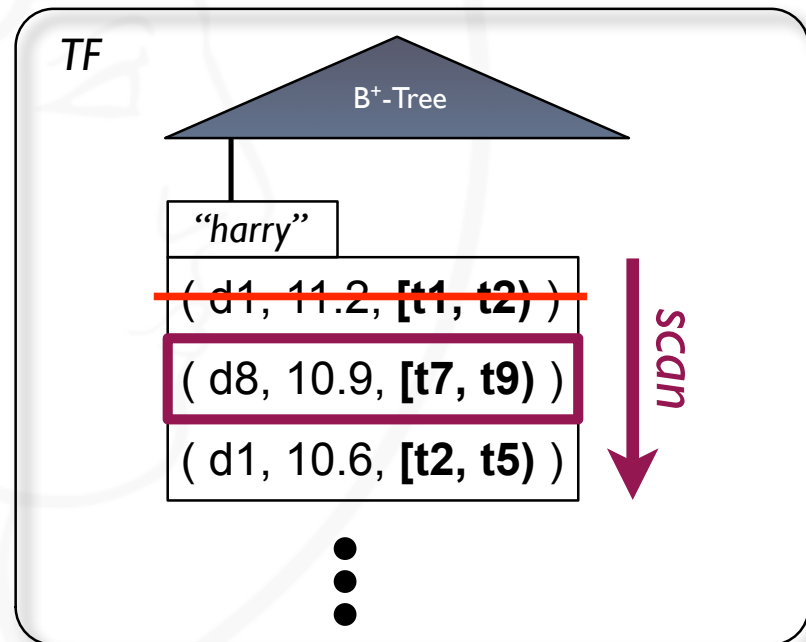


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

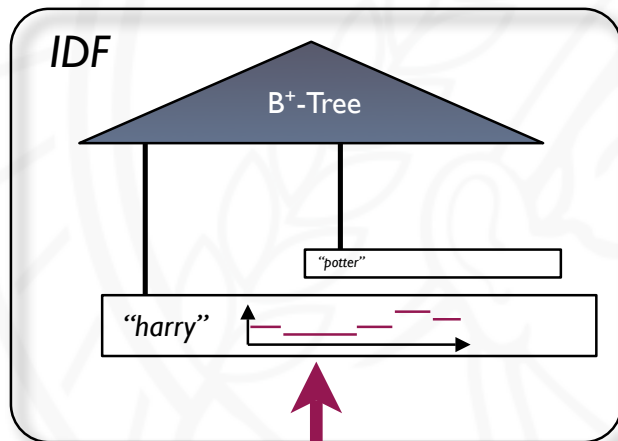


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

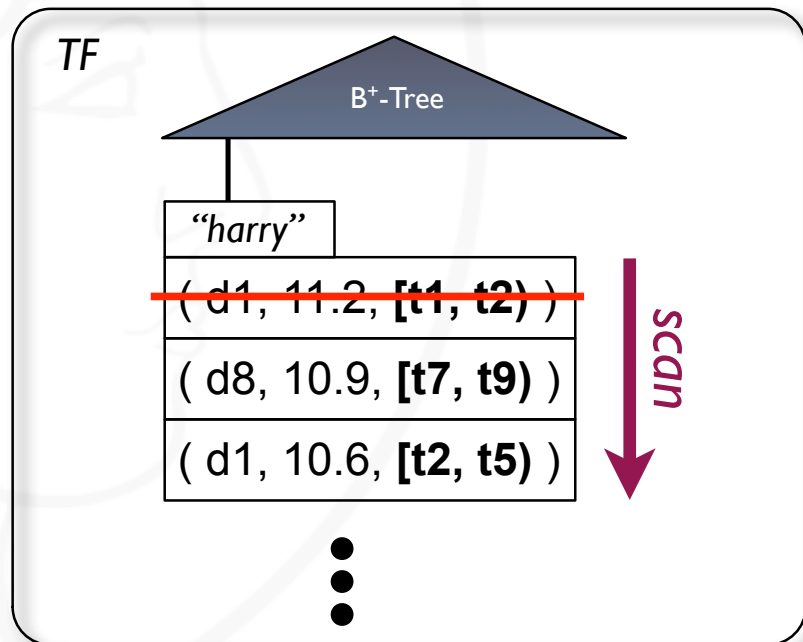


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: “harry”@ $t_8$

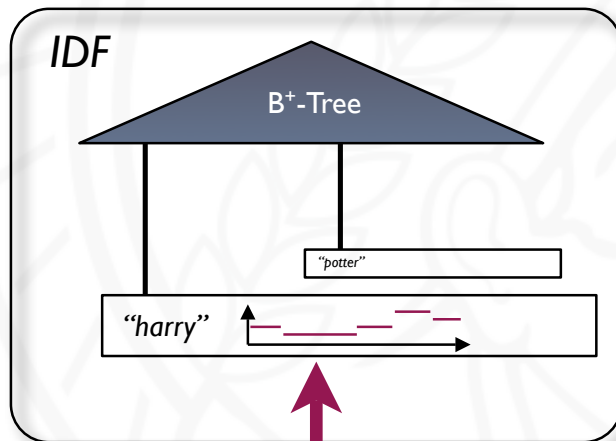


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

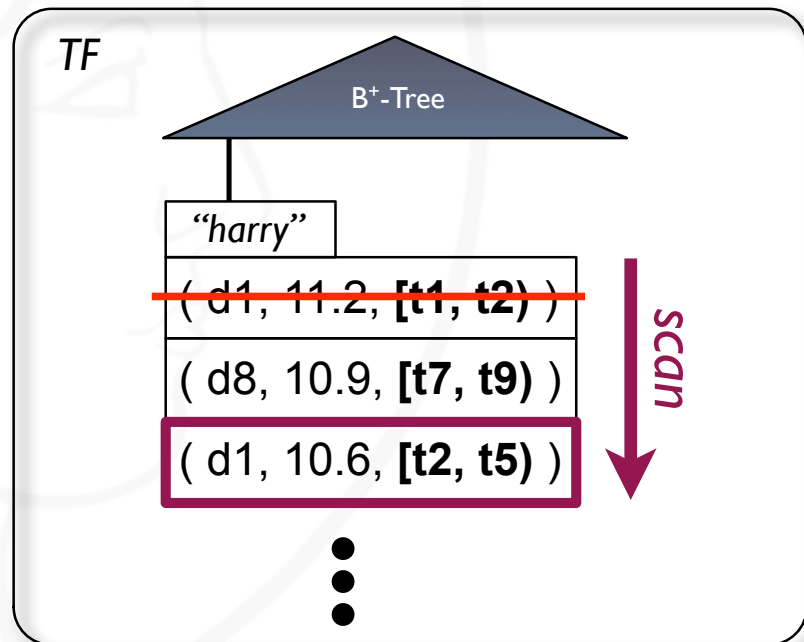


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: “harry”@ $t_8$

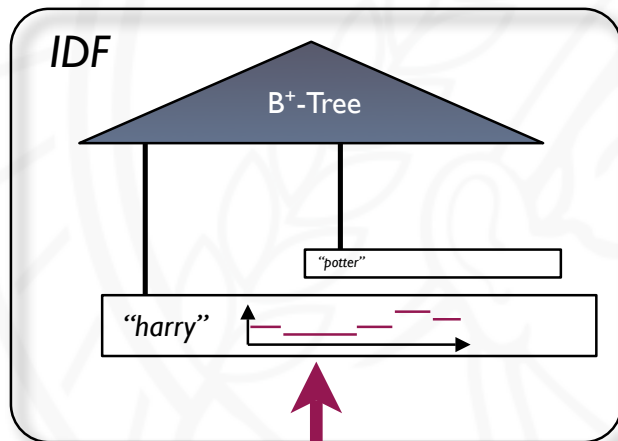


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

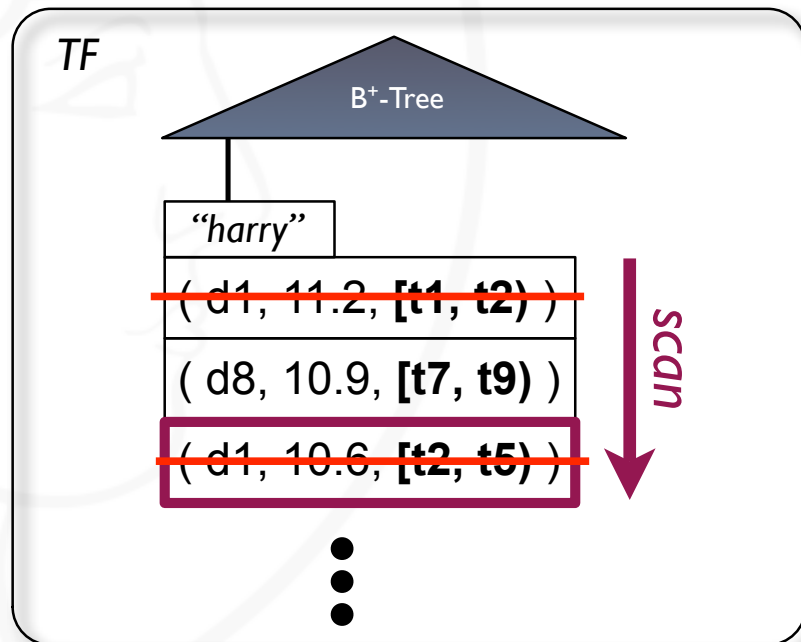


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*

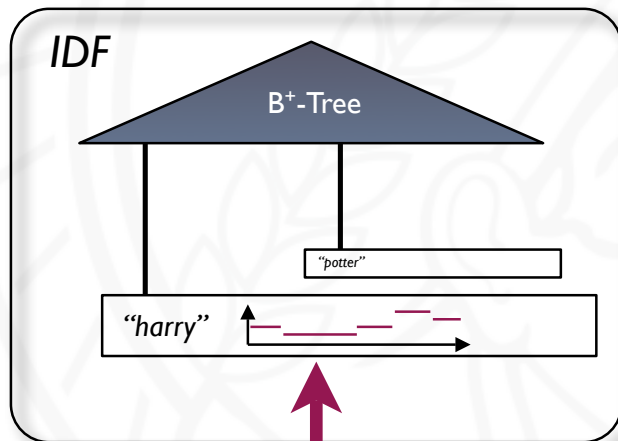


$$w_{idf}(\text{“harry”}, t_8) = 3.08$$

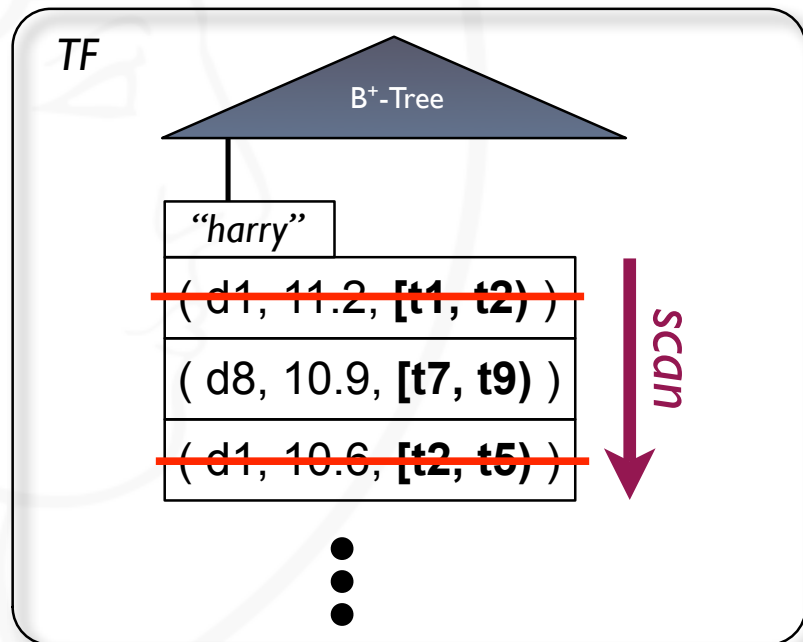


# Time-Travel Inverted File Index

- Time-travel query  $q^t$  can be processed by scanning index lists while ignoring non-relevant postings
- Example: *“harry”@t8*



$$w_{idf}(\text{“harry”}, t_8) = 3.08$$



# Outline

- Motivation
- Collection, Query, and Relevance Model
- Time-Travel Inverted File Index
  - Reducing Index Size
  - Tuning Index Performance
- Experimental Evaluation
- Conclusions

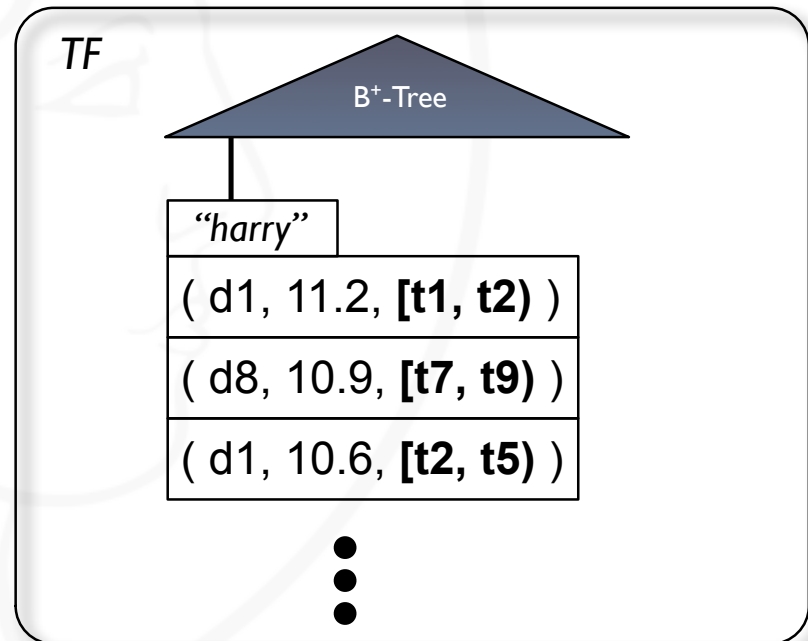


# Reducing Index Size

- Shortcoming: Since we create **one posting per version per term**, the resulting index is **very large**

**HUGE!!!**

(Wikipedia Revision  
History ~8.6B postings)



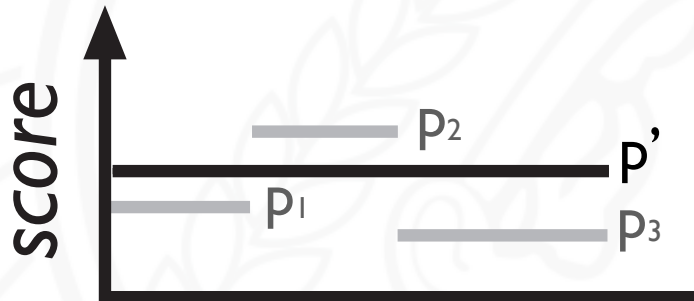
# Reducing Index Size

- Observation: Changes between document versions
  - **minor** (e.g., corrected typos)
  - have **no noticeable effect** on the ranked result (e.g., 500 x “harry” vs. 510 x “harry”)
- Idea: **Coalesce** sequences of temporally adjacent **postings** having **similar scores**



# Reducing Index Size

- Problem Statement: Given **input sequence  $I$**  find a **minimal length output sequence  $O$**  with approximation errors bounded by a threshold  $\epsilon$



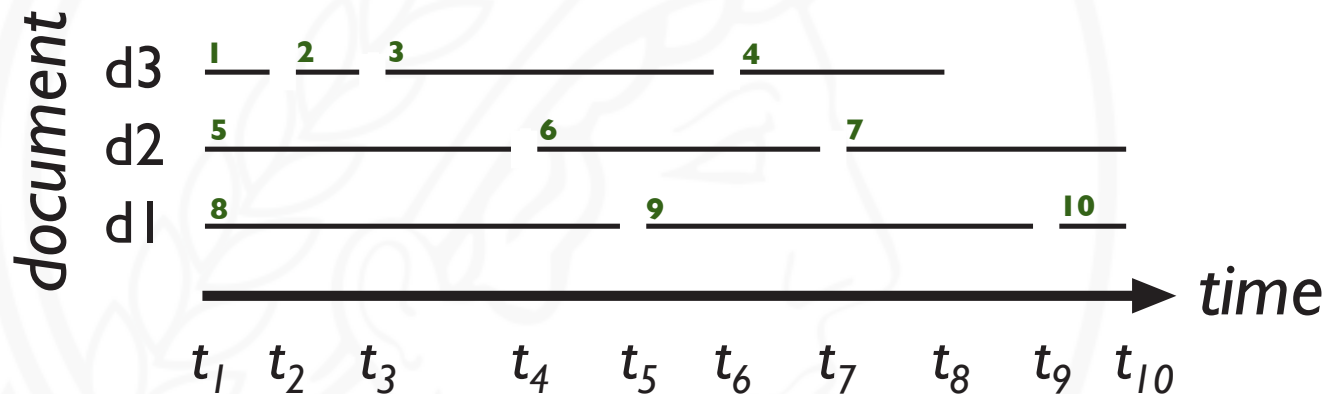
Guarantee:

$$|p' - p_i| / |p_i| \leq \epsilon$$

- **Approximate Temporal Coalescing (ATC)** finds an **optimal** output sequence using a greedy **linear time algorithm**

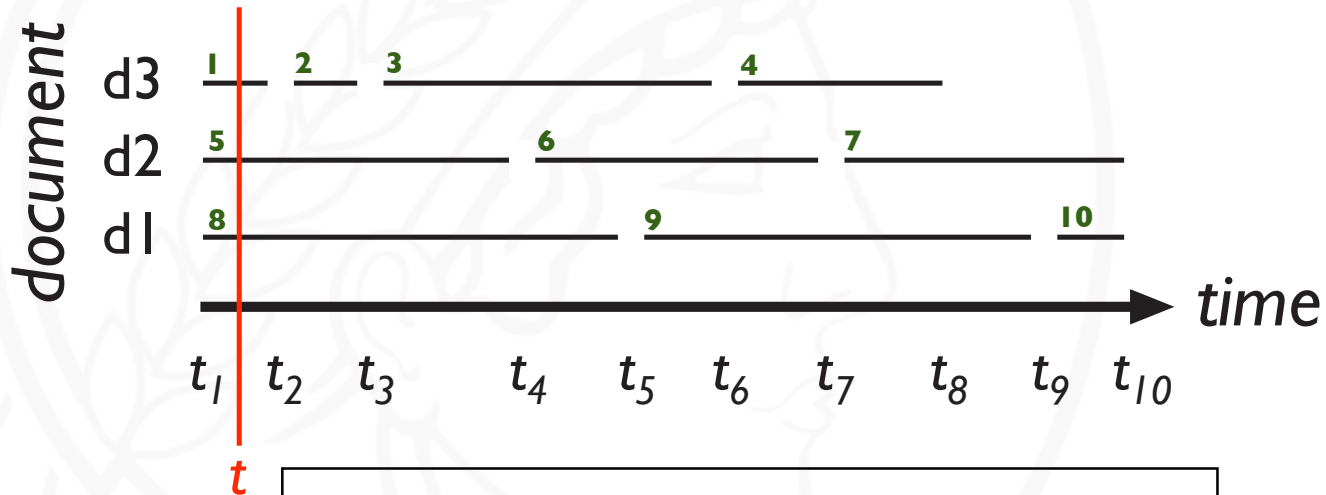
# Tuning Index Performance

- Shortcoming: During query processing **many postings** are **superfluously read**



# Tuning Index Performance

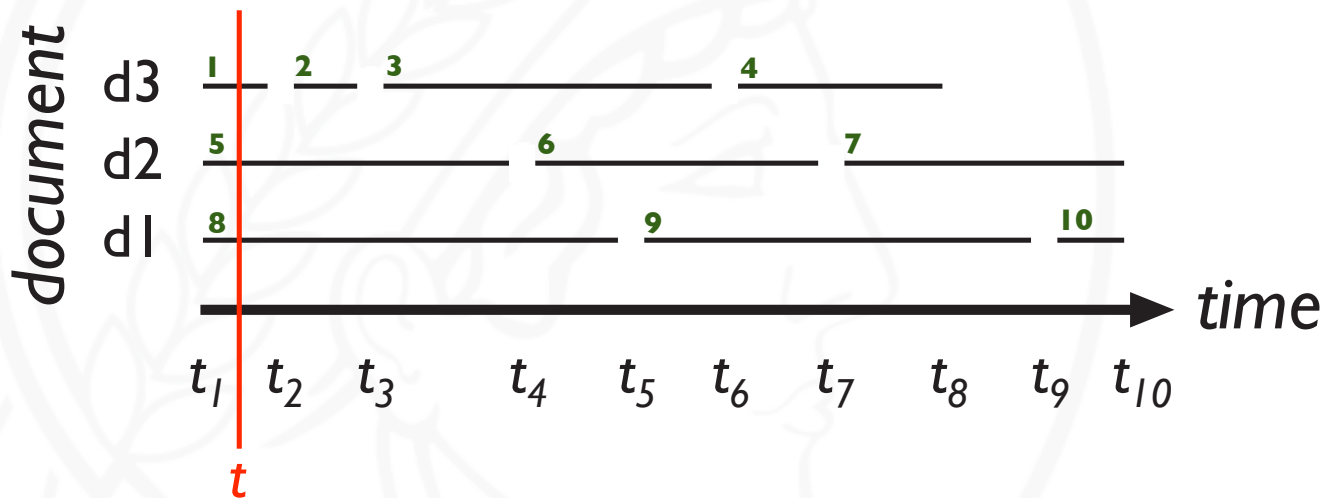
- Shortcoming: During query processing **many postings** are **superfluously read**



We read **10 postings**,  
but **only {1, 5, 8}** are **needed**

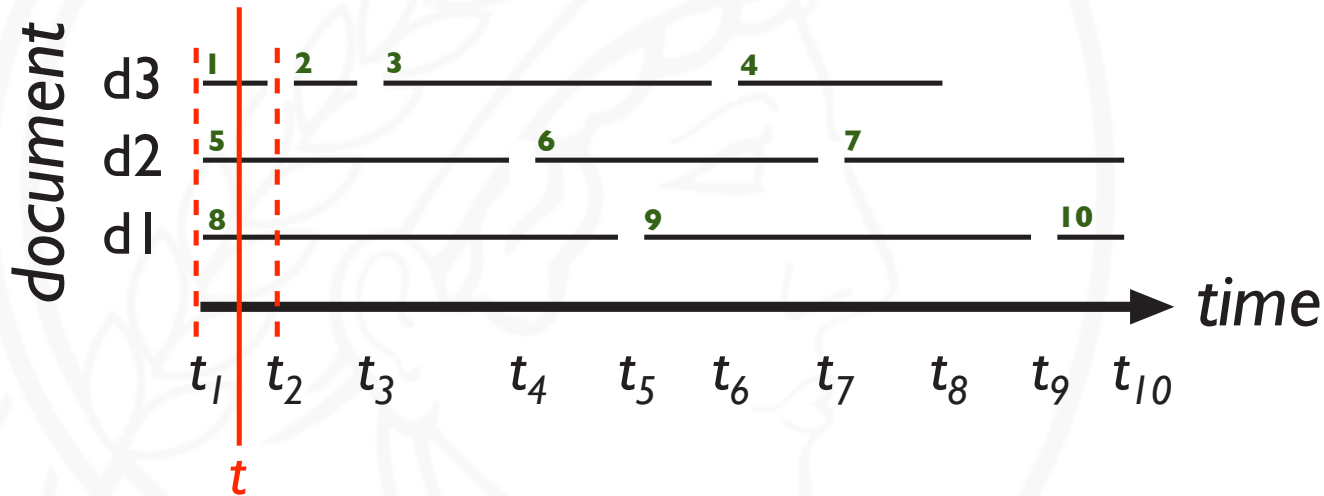
# Tuning Index Performance

- Idea: **Materialize smaller sublists** containing only postings that overlap with a smaller time-interval



# Tuning Index Performance

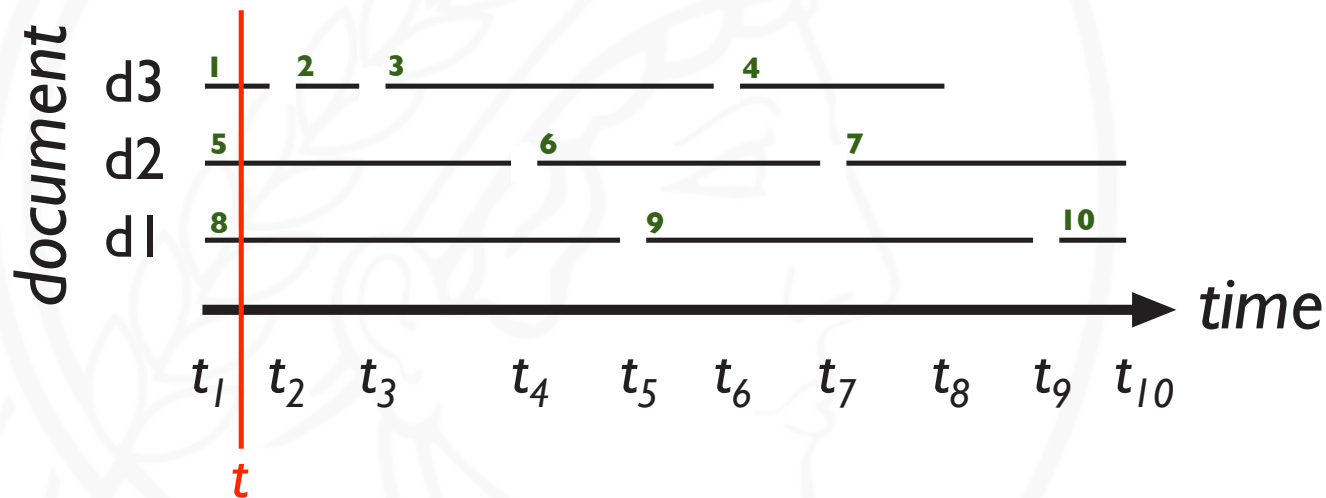
- Idea: **Materialize smaller sublists** containing only postings that overlap with a smaller time-interval



By materializing a **sublist for  $[t_1, t_2)$**  we can achieve **optimal performance** for the query

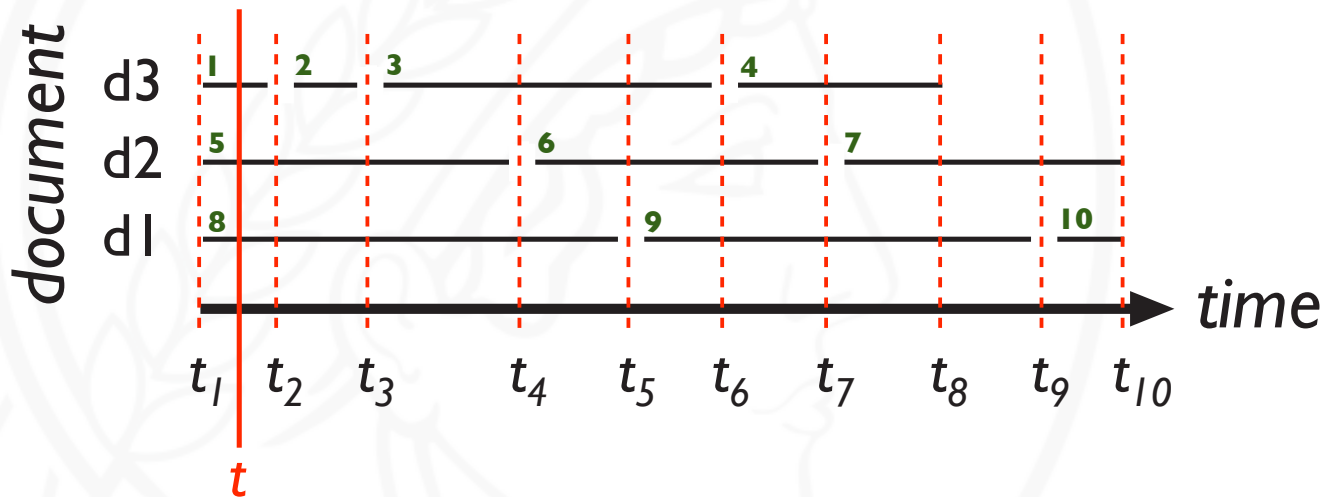
# Tuning Index Performance

- Idea: **Materialize smaller sublists** containing only postings that overlap with a smaller time-interval



# Tuning Index Performance

- Idea: **Materialize smaller sublists** containing only postings that overlap with a smaller time-interval



By materializing a sublist for **each elementary time interval** we achieve **optimal performance**

# Tuning Index Performance

- So far, we have seen **two extreme solutions**
  - **space-optimal**: keep only a single list ( $S_{OPT}$ )
  - **performance-optimal**: keep one list per elementary time-interval ( $P_{OPT}$ )
- We propose two systematic techniques to trade-off space and performance
  - **performance-guarantee**: consumes minimal space while retaining a performance guarantee (PG)
  - **space-bound**: achieves best performance while not exceeding a space limit (SB)



# Tuning Index Performance

- Performance Guarantee (PG)
  - consumes **minimal space**
  - guarantees that for any  $t$  at most  $\gamma \cdot n_t$  **postings are read** where  $n_t$  is the **number of postings that exist at time  $t$**
- Optimal solution computable by means of induction in  **$O(T^2)$  time** and  **$O(T^2)$  space** (where  $T$  is the number of distinct timestamps in the list)

# Tuning Index Performance

- Space Bound (SB)
  - achieves **minimal expected processing cost** (i.e., expected length of the list that is scanned)
  - consumes **at most  $K \cdot n$  space** where  $n$  is the length of the original list
- Optimal solution computable using dynamic programming in  **$O(n^4)$  time** and  **$O(n^3)$  space**
- Approximate solution computable in  **$O(T^2)$  time** and  **$O(T)$  space** using **simulated annealing**



# Outline

- Motivation
- Collection, Query, and Relevance Model
- Time-Travel Inverted File Index
  - Reducing Index Size
  - Tuning Index Performance
- **Experimental Evaluation**
- **Conclusions**



# Experimental Evaluation – Setup

## ■ Implementation:

- Java, Oracle 10g

## ■ Datasets:

- **WIKI:** Revision history of [English Wikipedia](#) (2001-2005)  
892K documents / 13,976K versions / 0.7 TBytes
- **UKGOV:** Weekly crawls of [11 .gov.uk sites](#) (2004-2005)  
502K documents / 8,687K versions / 0.4 TBytes

## ■ Queries:

- [300 keyword queries](#) from AOL query log that most frequently produced a result click on en.wikipedia.org / .gov.uk
- Each keyword query is assigned [one time point per month](#) in the collection's lifespan (18K / 7.2K time-travel queries in total)



# Experimental Evaluation – Setup

- Implementation:
  - Java, Oracle 10g
- Datasets:
  - **WIKI:** Revision history of [English Wikipedia](#) (2001-2005)  
892K documents / 13,976K versions / 0.7 TBytes

WIKI: UKGOV: Weekly crawls of 11 .gov.uk sites (2004-2005)  
502K documents / 8,697K versions / 0.4 TBytes  
*ten commandments, abraham lincoln, da vinci code, harlem renaissance...*

## Queries:

UKGOV: 100 keyword queries from AOL query log that most frequently  
*1901 uk census, british royal family, migrant worker statistics, witness intimidation...*  
Each keyword query is assigned one time point per month in the collection's lifespan (18K / 7.2K time-travel queries in total)



# Experimental Evaluation – Setup

## ■ Implementation:

- Java, Oracle 10g

## ■ Datasets:

- **WIKI:** Revision history of [English Wikipedia](#) (2001-2005)  
892K documents / 13,976K versions / 0.7 TBytes
- **UKGOV:** Weekly crawls of [11 .gov.uk sites](#) (2004-2005)  
502K documents / 8,687K versions / 0.4 TBytes

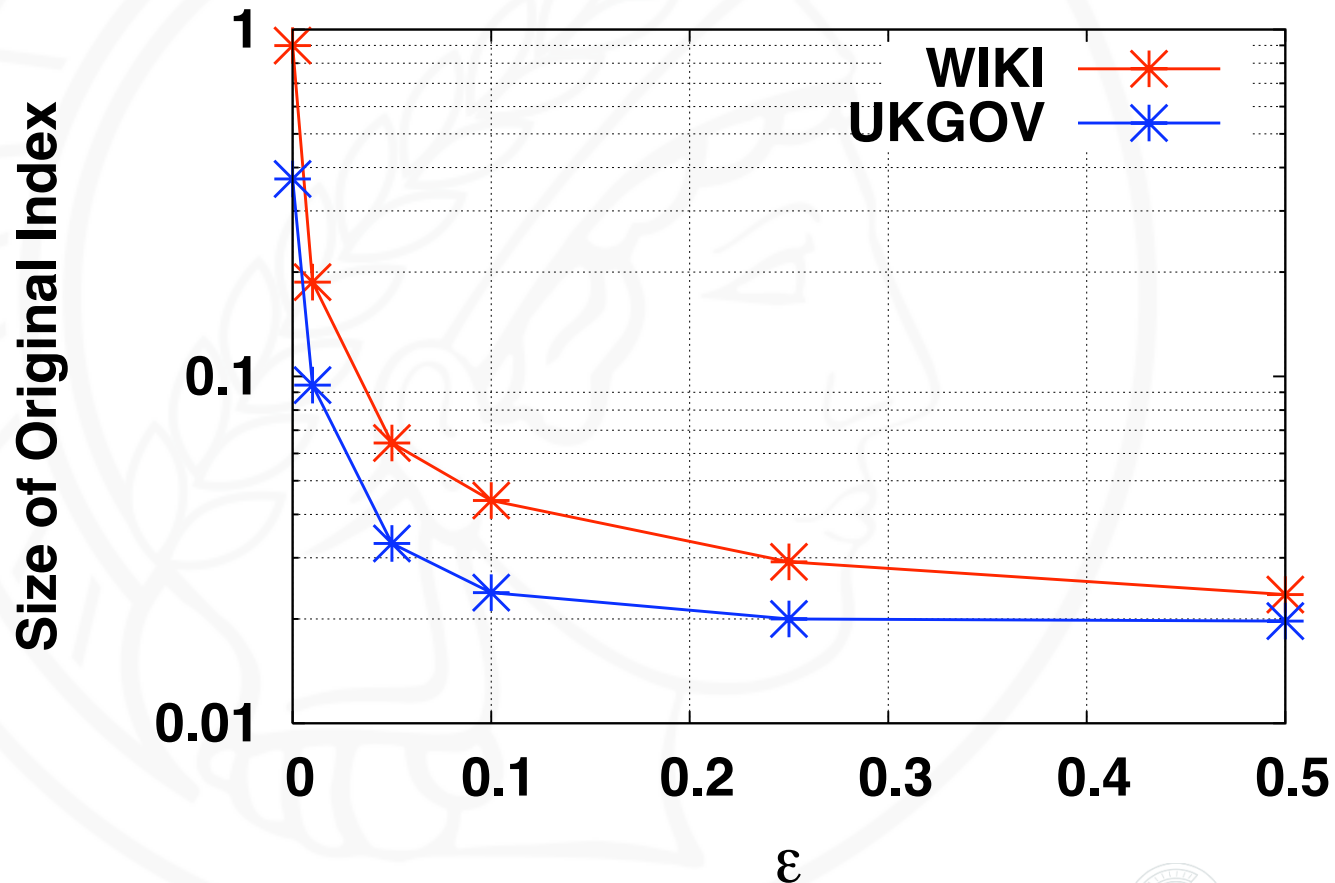
## ■ Queries:

- [300 keyword queries](#) from AOL query log that most frequently produced a result click on en.wikipedia.org / .gov.uk
- Each keyword query is assigned [one time point per month](#) in the collection's lifespan (18K / 7.2K time-travel queries in total)



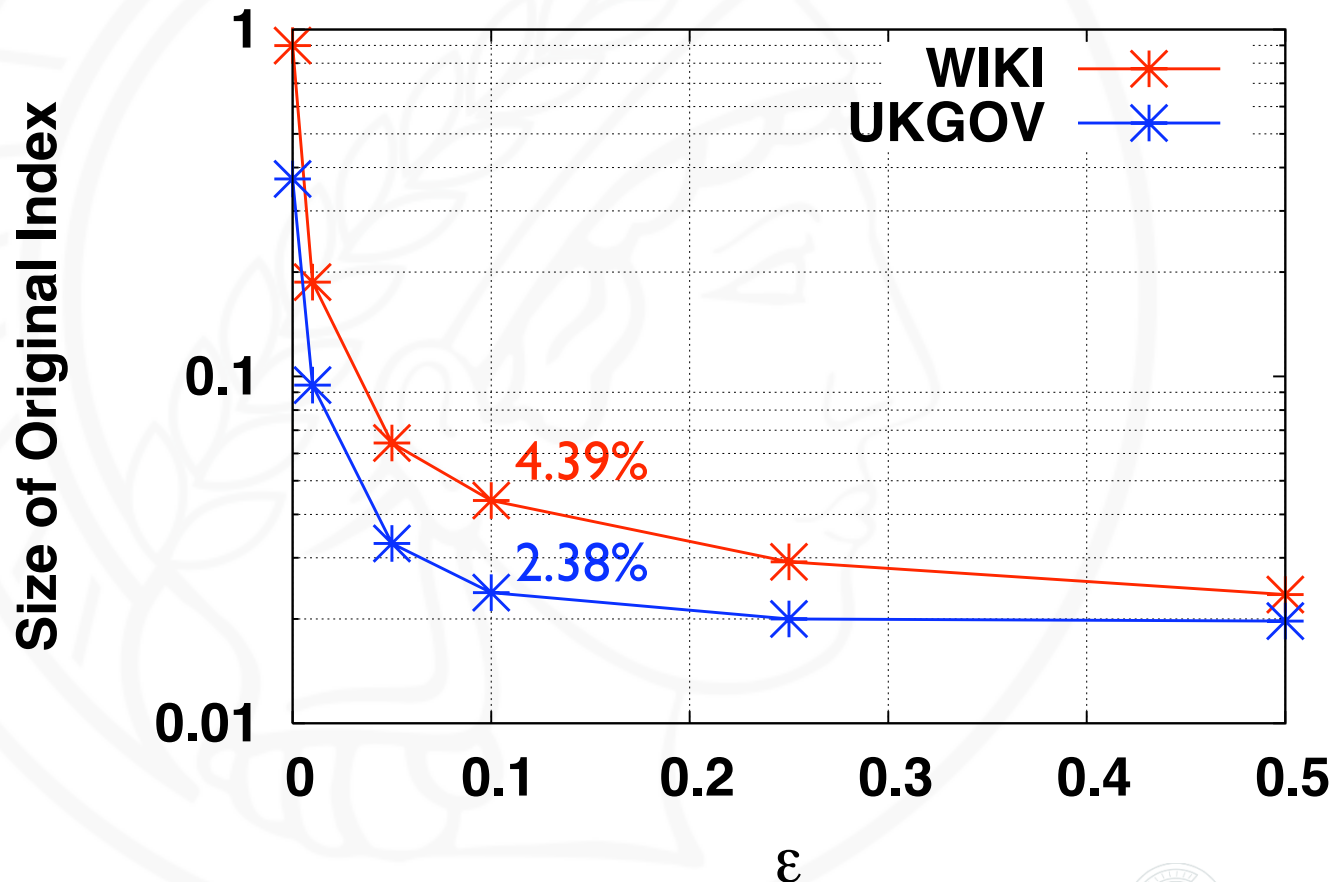
# Approximate Temporal Coalescing

- Indexes computed for different values of threshold  $\epsilon$



# Approximate Temporal Coalescing

- Indexes computed for different values of threshold  $\epsilon$

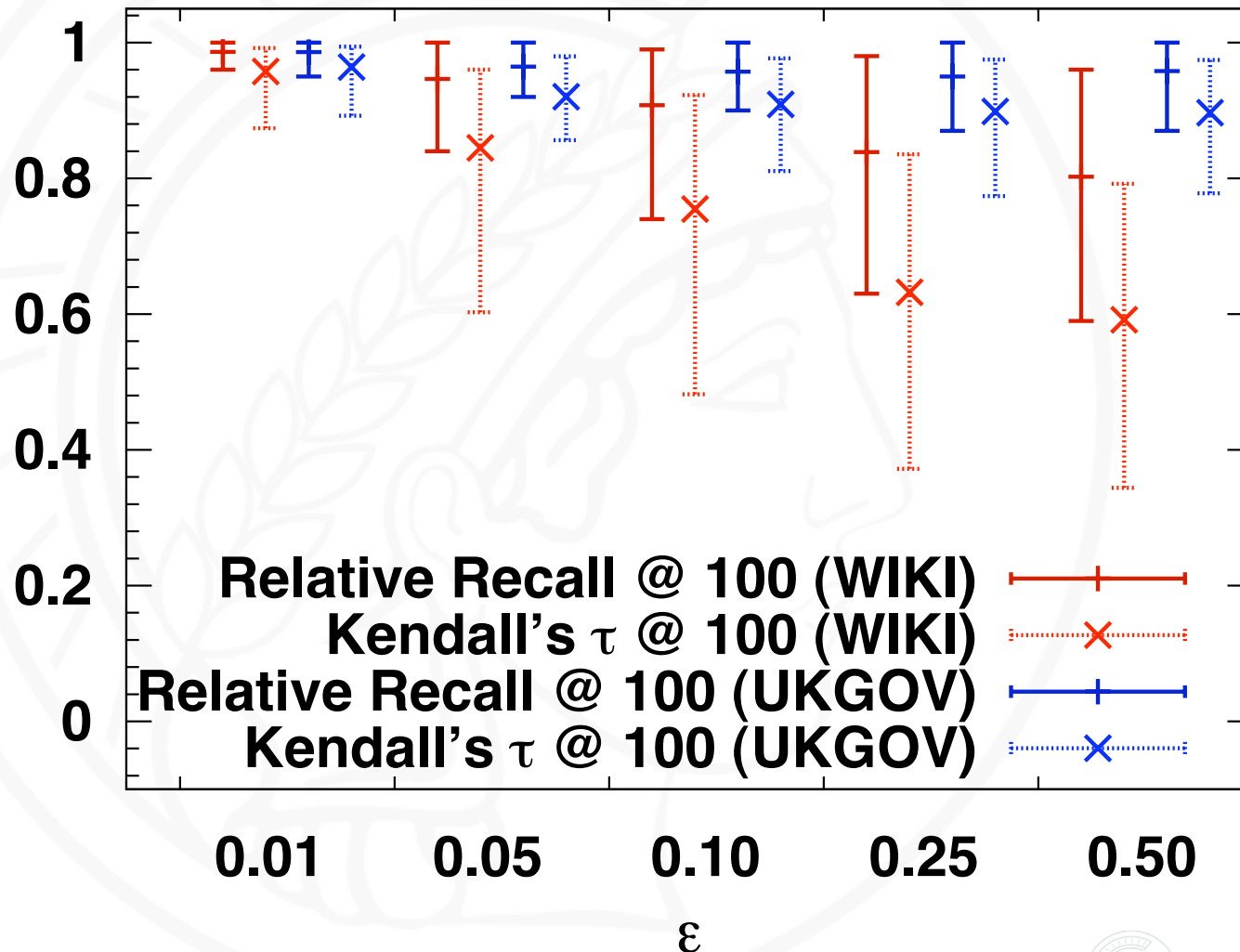


# Approximate Temporal Coalescing

- Impact on top- $k$  query results assessed using
  - **Relative Recall** @  $k$  in  $[0, 1]$
  - **Kendall's  $\tau$**  @  $k$  in  $[-1, 1]$
- Computed per dataset for
  - all time-travel queries (18K / 7.2K)
  - $k$  varying as 10, 25, 50, 100
  - $\epsilon$  varying as 0.01, 0.05, 0.10, 0.25, 0.50
- We report mean, 5%-percentile, and 95%-percentile



# Approximate Temporal Coalescing



# Sublist Materialization

- Threshold for ATC fixed as  $\epsilon = 0.10$
- For terms in query workloads (422/522) we apply
  - $S_{OPT}$  and  $P_{OPT}$
  - PG for  $\gamma$  varying between 1.10 and 3.00
  - SB for  $K$  varying between 1.10 and 3.00
- We report
  - **Space**, i.e., total number of postings in materialized sublists
  - **Expected Processing Cost (EPC)**, i.e., expected length of scanned list for random term and time



# Performance Guarantee

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Performance Guarantee

|                 | WIKI   |      | UKGOV |      |
|-----------------|--------|------|-------|------|
|                 | Space  | EPC  | Space | EPC  |
| $\gamma = 1.10$ | 1,004% | 106% | 616%  | 103% |
| $\gamma = 1.50$ | 295%   | 132% | 233%  | 117% |
| $\gamma = 2.00$ | 195%   | 160% | 163%  | 125% |
| $\gamma = 3.00$ | 145%   | 207% | 132%  | 133% |

EPC = Expected Processing Cost



# Performance Guarantee

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Performance Guarantee

|                 | WIKI   |      | UKGOV |      |
|-----------------|--------|------|-------|------|
|                 | Space  | EPC  | Space | EPC  |
| $\gamma = 1.10$ | 1,004% | 106% | 616%  | 103% |
| $\gamma = 1.50$ | 295%   | 132% | 233%  | 117% |
| $\gamma = 2.00$ | 195%   | 160% | 163%  | 125% |
| $\gamma = 3.00$ | 145%   | 207% | 132%  | 133% |

EPC = Expected Processing Cost



# Performance Guarantee

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Performance Guarantee

|                 | WIKI   |      | UKGOV |      |
|-----------------|--------|------|-------|------|
|                 | Space  | EPC  | Space | EPC  |
| $\gamma = 1.10$ | 1,004% | 106% | 616%  | 103% |
| $\gamma = 1.50$ | 295%   | 132% | 233%  | 117% |
| $\gamma = 2.00$ | 195%   | 160% | 163%  | 125% |
| $\gamma = 3.00$ | 145%   | 207% | 132%  | 133% |

EPC = Expected Processing Cost



# Performance Guarantee

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Performance Guarantee

|                 | WIKI   |      | UKGOV |      |
|-----------------|--------|------|-------|------|
|                 | Space  | EPC  | Space | EPC  |
| $\gamma = 1.10$ | 1,004% | 106% | 616%  | 103% |
| $\gamma = 1.50$ | 295%   | 132% | 233%  | 117% |
| $\gamma = 2.00$ | 195%   | 160% | 163%  | 125% |
| $\gamma = 3.00$ | 145%   | 207% | 132%  | 133% |

EPC = Expected Processing Cost



# Performance Guarantee

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Performance Guarantee

|                 | WIKI   |      | UKGOV |      |
|-----------------|--------|------|-------|------|
|                 | Space  | EPC  | Space | EPC  |
| $\gamma = 1.10$ | 1,004% | 106% | 616%  | 103% |
| $\gamma = 1.50$ | 295%   | 132% | 233%  | 117% |
| $\gamma = 2.00$ | 195%   | 160% | 163%  | 125% |
| $\gamma = 3.00$ | 145%   | 207% | 132%  | 133% |

EPC = Expected Processing Cost



# Space Bound

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Space Bound

|                 | WIKI  |      | UKGOV |      |
|-----------------|-------|------|-------|------|
|                 | Space | EPC  | Space | EPC  |
| $\kappa = 3.00$ | 288%  | 139% | 273%  | 107% |
| $\kappa = 2.00$ | 194%  | 171% | 180%  | 119% |
| $\kappa = 1.50$ | 146%  | 214% | 131%  | 131% |
| $\kappa = 1.10$ | 109%  | 406% | 104%  | 145% |

EPC = Expected Processing Cost



# Space Bound

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Space Bound

|                 | WIKI  |      | UKGOV |      |
|-----------------|-------|------|-------|------|
|                 | Space | EPC  | Space | EPC  |
| $\kappa = 3.00$ | 288%  | 139% | 273%  | 107% |
| $\kappa = 2.00$ | 194%  | 171% | 180%  | 119% |
| $\kappa = 1.50$ | 146%  | 214% | 131%  | 131% |
| $\kappa = 1.10$ | 109%  | 406% | 104%  | 145% |

EPC = Expected Processing Cost



# Space Bound

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Space Bound

|                 | WIKI  |      | UKGOV |      |
|-----------------|-------|------|-------|------|
|                 | Space | EPC  | Space | EPC  |
| $\kappa = 3.00$ | 288%  | 139% | 273%  | 107% |
| $\kappa = 2.00$ | 194%  | 171% | 180%  | 119% |
| $\kappa = 1.50$ | 146%  | 214% | 131%  | 131% |
| $\kappa = 1.10$ | 109%  | 406% | 104%  | 145% |

EPC = Expected Processing Cost



# Space Bound

|                  | WIKI    |      | UKGOV   |      |
|------------------|---------|------|---------|------|
|                  | Space   | EPC  | Space   | EPC  |
| P <sub>OPT</sub> | 14,428% | 100% | 11,406% | 100% |
| S <sub>OPT</sub> | 100%    | 963% | 100%    | 147% |

## Space Bound

|                 | WIKI  |      | UKGOV |      |
|-----------------|-------|------|-------|------|
|                 | Space | EPC  | Space | EPC  |
| $\kappa = 3.00$ | 288%  | 139% | 273%  | 107% |
| $\kappa = 2.00$ | 194%  | 171% | 180%  | 119% |
| $\kappa = 1.50$ | 146%  | 214% | 131%  | 131% |
| $\kappa = 1.10$ | 109%  | 406% | 104%  | 145% |

EPC = Expected Processing Cost



# Space Bound

|           | WIKI    |      | UKGOV   |      |
|-----------|---------|------|---------|------|
|           | Space   | EPC  | Space   | EPC  |
| $P_{OPT}$ | 14,428% | 100% | 11,406% | 100% |
| $S_{OPT}$ | 100%    | 963% | 100%    | 147% |

## Space Bound

|                 | WIKI  |      | UKGOV |      |
|-----------------|-------|------|-------|------|
|                 | Space | EPC  | Space | EPC  |
| $\kappa = 3.00$ | 288%  | 139% | 273%  | 107% |
| $\kappa = 2.00$ | 194%  | 171% | 180%  | 119% |
| $\kappa = 1.50$ | 146%  | 214% | 131%  | 131% |
| $\kappa = 1.10$ | 109%  | 406% | 104%  | 145% |

EPC = Expected Processing Cost



# Outline

- Motivation
- Collection, Query, and Relevance Model
- Time-Travel Inverted File Index
  - Reducing Index Size
  - Tuning Index Performance
- Experimental Evaluation
- **Conclusions**



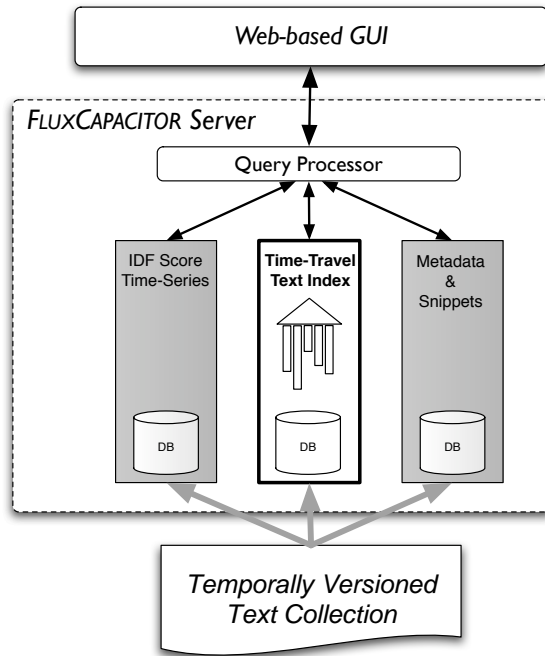
# Conclusions

- Time-Travel Text Search
  - an interesting & important research problem!
- Our Time Machine
  - building on inverted file index
  - significant reduction of index size
  - tunable index performance
- Experimental Evaluation
  - over two large-scale real-world datasets
  - demonstrating efficiency & effectiveness



# Demo at VLDB '07

September 23-28, Vienna, Austria



## FluxCapacitor

Your query **iraq war** @ Jun 18, 2002 8:30 PM needed 266ms and has 50 results

[Iran-Iraq\\_War](#)  
The b Iran Iraq War b was a border war between Iran and Iraq that took place between September 22 1980 and August 20 1988 It is also known as the b First Persian Gulf War b and the b Gulf War b  
**Score:** 13,385 | **Created:** May 27, 2002 5:59 PM

[Gulf\\_War](#)  
The b Gulf War b also known as b Persian Gulf War b b War in the Gulf b b Iraq Kuwait Conflict b b Second Gulf War b or b UN Iraq conflict b was a conflict between Iraq and a coalition force led  
**Score:** 13,343 | **Created:** Jun 18, 2002 10:10 AM

[History\\_of\\_Iraq](#)  
Ancient Times For most of historic time the city and empire of Babylon occupied parts of the present time region of Iraq There were many dynasties and kingdoms which ruled Babylon and other  
**Score:** 12,76 | **Created:** Jun 10, 2002 3:01 AM





# Thank you!

# Questions?



# Experimental Evaluation – PG & SB

