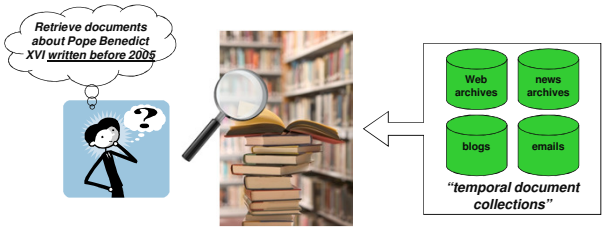
Web Dynamics

Analysis and Implications from Search Perspective

Motivation

- Searching documents created/edited over time
  - E.g., web archives, news archives, blogs, or emails



Term-based IR approaches may give unsatisfied results

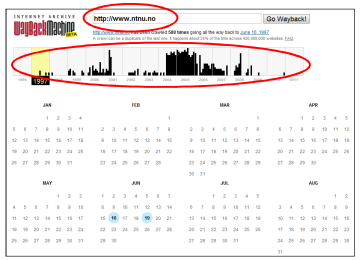
Lecturers

Dr. Ismail Sengör Altingövde
- Senior researcher
- L3S Research Center
- Hannover, Germany

Dr. Nattiya Kanhabua
- Postdoc
- L3S Research Center
- Hannover, Germany

Wayback Machine[1]

- A web **archive search** tool by the *Internet Archive*
  - Query by a URL, e.g., *http://www.ntnu.no*



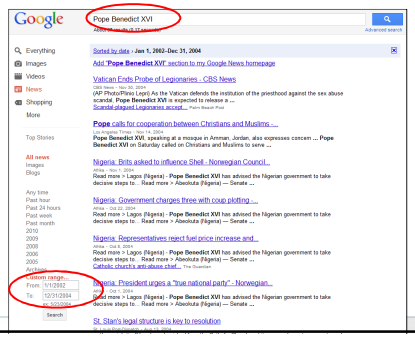
No keyword query

No relevance ranking

[1]Retrieved on 15 January 2011

Google News Archive Search

- A **news archive search** tool by Google
  - Query by keywords
  - Rank results by relevance or date



Not consider terminology changes over time

Evolution of the Web

- Web is *changing* over time in many aspects:
  - **Size**: web pages are added/deleted all the time
  - **Content**: web pages are edited/modified
  - **Query**: users' information needs changes, entity-relationship changes over time
  - **Usage**: users' behaviors change over time

Outline

Day 1: Introduction
- Evolution of the Web
- Overview of research topics
- Content and query analysis

Day 2: Evolution of Web search results
- Short-term impacts
- Longitudinal analysis

Day 3: Indexing the Past
- Indexing and searching versioned documents

Day 4: Retrieval and ranking
- Searching the past
- Searching the future

Size dynamics

- Challenges
  - Crawling, indexing, and caching

Content dynamics

- Challenges
  - Document representation and retrieval

Behavior dynamics

- Challenges
  - Browsing and search behavior

Query dynamics

- Challenges
  - Query understanding and representations
  - Time-sensitive queries

Research topics

- Content analysis
  - Determining timestamps of documents
  - Temporal information extraction
- Query analysis
  - Determining time of queries
  - Named entity evolution
  - Query performance prediction
- Evolution of Search Results
  - Short-term impacts on result caches
  - Longitudinal analysis of search results

Research topics (cont')

- Indexing
  - Indexing and query processing techniques for the versioned document collections
- Retrieval and ranking
  - Searching the past
  - Searching the future

Motivation

- Incorporating the time dimension into search can increase retrieval effectiveness
  - *Only if temporal information is available*
- Research question
  - How to determine the temporal information of documents?

Content Analysis

(1) Determining timestamps of documents

(2) Temporal information extraction

Two time aspects

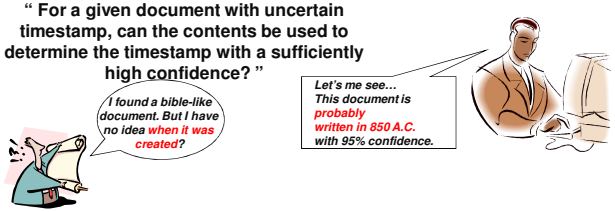
1. Publication or modified time
   - ***Determining timestamps of documents***
   - Meta-data generation
2. Content or event time
   - ***Temporal information extraction***
   - Natural language processing

Determining time of documents

Problem Statements

- Difficult to find the *trustworthy* time for web documents
  - Time gap between crawling and indexing
  - Decentralization and relocation of web documents
  - No standard metadata for time/date

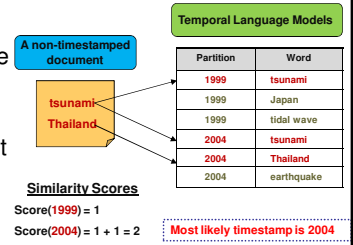
" For a given document with uncertain timestamp, can the contents be used to determine the timestamp with a sufficiently high confidence? "



Content-based approach

Temporal Language Models

- Based on the statistic usage of words over time
- Compare each word of a non-timestamped document with a reference corpus
- Tentative timestamp -- a time partition mostly overlaps in word usage



[de Jong et al., AHC 2005]

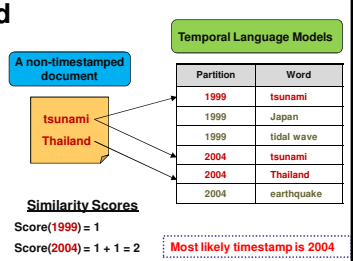
Current approaches

1. Content-based
2. Link-based
3. Hybrid

Normalized log-likelihood ratio

Normalized log-likelihood ratio

- Variant of Kullback-Leibler divergence
- Similarity of a document and time partitions
- C is the background model estimated on the corpus
- Linear interpolation smoothing to avoid the zero probability of unseen words



$$Score(d_i, p_j) = \sum_{w \in d_i} P(w|d_i) \times \log \frac{P(w|p_j)}{P(w|C)}$$

[Kraaij, SIGIR Forum 2005]

Improving temporal LMs

- **Enhancement techniques**
  1. Semantic-based data preprocessing
  2. Search statistics to enhance similarity scores
  3. Temporal entropy as term weights

Intuition: A term weight depends on how good the term is for separating time partitions (**discriminative**)

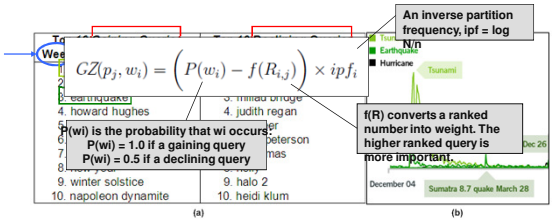
Approach: Propose **temporal entropy**, based on a term selection presented in Lochbaum and Streeter

[Kanhabua et al., ECDL 2008]

Leveraging search statistics

Intuition: Search statistics *Google Zeitgeist* (GZ) can increase the probability of a tentative time partition

Approach: Linearly combine a *GZ* score with the normalized log-likelihood ratio



[Kanhabua et al., ECDL 2008]

Semantic-based preprocessing

Intuition: *Direct comparison* between extracted words and corpus partitions has *limited accuracy*

Approach: Integrate *semantic-based* techniques into document preprocessing

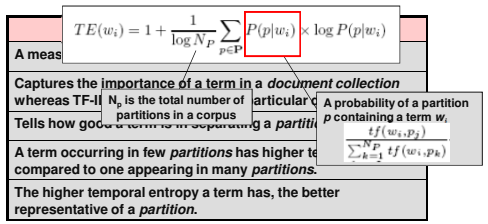
Semantic-based Preprocessing	Description
Part-of-speech tagging	Select only interesting classes of words, e.g. nouns, verbs, and adjectives
Collocation extraction	Co-occurrence of different words can alter the meaning, e.g. "United States"
Word sense disambiguation	Identify the correct sense of a word from context, e.g. "bank"
Concept extraction	Compare concepts instead of original words, e.g. "tsunami" and "tidal wave" have the common concept of "disaster"
Word filtering	Select the top-ranked words according to TF-IDF scores for a comparison

[Kanhabua et al., ECDL 2008]

Temporal entropy

Intuition: A term weight depends on how good the tern is for separating time partitions (*discriminative*)

Approach: Propose *temporal entropy*, based on a term selection presented in Lochbaum and Streeter



[Kanhabua et al., ECDL 2008]

Link-based approach

- **Dating a document using its neighbors**
  1. Web pages linking to the document
     - Incoming links
  2. Web pages pointed by the document
     - Outgoing links
  3. Media assets associated with the document
     - E.g., images
- Averaging the last-modified dates of its neighbors as *timestamps*

[Nunes et al., WIDM 2007]

Temporal information extraction

- Extract *temporal expressions* using time and event recognition algorithms
- Three types of temporal expressions
  1. **Explicit**: time mentions being mapped directly to a time point or interval, e.g., “July 4, 2012”
  2. **Implicit**: imprecise time point or interval, e.g., “Independence Day 2012”
  3. **Relative:** resolved to a time point or interval using other types or the publication date, e.g., “next month”

[Alonso et al., SIGIR Forum 2007; Verhagen et al., ACL 2005]
[Strötgen et al., SemEval 2010]

Hybrid approach

- **Inferring timestamps using machine learning**
  - Exploit links, contents of a web pages and its neighbors
  - Features: linguistic, position, page formats, and tags

[Chen et al., SIGIR 2010]

References

- [Alonso et al., SIGIR Forum 2007] Omar Alonso, Michael Gertz, Ricardo A. Baeza-Yates: On the value of temporal information in information retrieval. SIGIR Forum 41(2): 35-41 (2007)
- [Chen et al., SIGIR 2010] Zhumin Chen, Jun Ma, Chaoran Cui, Hongxing Rui, Shaomang Huang: Web page publication time detection and its application for page rank. SIGIR 2010: 859-860
- [de Jong et al., AHC 2005] Franciska de Jong, Henning Rode, Djoerd Hiemstra: Temporal language models for the disclosure of historical text. AHC 2005: 161-168
- [Kanhabua et al., ECDL 2008] Nattiya Kanhabua, Kjetil Nørvåg: Improving Temporal Language Models for Determining Time of Non-timestamped Documents. ECDL 2008: 358-370
- [Kraaij, SIGIR Forum 2005] Wessel Kraaij: Variations on language modeling for information retrieval. SIGIR Forum 39(1): 61 (2005)
- [Nunes et al., WIDM 2007] Sérgio Nunes, Cristina Ribeiro, Gabriel David: Using neighbors to date web documents. WIDM 2007: 129-136
- [Strötgen et al., SemEval 2010] Jannik Strötgen, Michael Gertz: Heideltime: High quality rule-based extraction and normalization of temporal expressions. SemEval 2010: 321-324
- [Verhagen et al., ACL 2005] Marc Verhagen, Inderjeet Mani, Roser Sauri, Jessica Littman, Robert Knippen, Seok Bae Jang, Anna Rumshisky, John Phillips, James Pustejovsky: Automating Temporal Annotation with TARSQI. ACL 2005

Question?

Temporal queries

- Temporal information needs
  - Searching *temporal document collections,* .e.g, digital libraries, web/news archives
  - Historians, librarians, journalists or students
- Temporal queries exist in both standard collections and the Web
  - Relevancy is dependent on time
  - Documents are about events at particular time

[Berberich et al., ECIR 2010]

Query Analysis

(1) Determining time of queries
(2) Named entity evolution
(3) Query performance prediction

Distribution over time of Qrel

Recency query

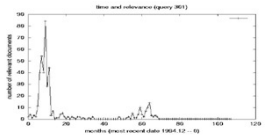


Figure 2.2: Query 301 "International Organized Crime" – A "recency" query.

Time-sensitive query

Figure 2.3: Query 156 "Efforts to Enact Gun Control Legislation"- Relevant documents mostly in the past.

Time-insensitive query

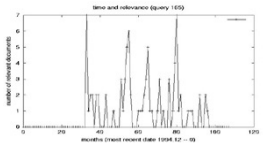


Figure 2.4: Query 165 "Tobacco Company Advertising and the Young" - More uniform distribution

[Li et al., CIKM 2003]

Types of temporal queries

- Two types of temporal queries
  1. **Explicit**: time is provided, "Presidential election 2012"
  2. **Implicit**: time is *not* provided, "Germany World Cup"
     - Temporal intent can be implicitly inferred
     - I.e., refer to the World Cup event in 2006
- Studies of web search query logs show a significant fraction of temporal queries
  - 1.5% of web queries are *explicit*
  - ~7% of web queries are *implicit*
  - 13.8% of queries contain *explicit* time and 17.1% of queries have temporal intent *implicitly* provided

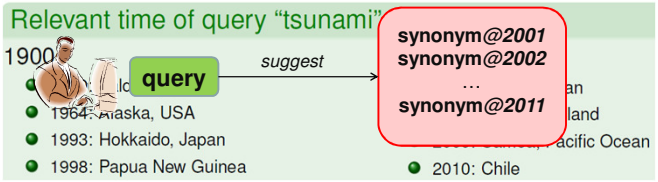
[Nunes et al., ECIR 2008; Metzler et al., SIGIR 2009; Zhang et al., EMNLP 2010]

Determining time of queries

- Problem statements
  - ***Implicit temporal queries***: users have no knowledge about the relevant time of a query
  - Difficult to achieve high accuracy using only keywords
  - Relevant results associated to particular time *not given*
- Research question
  - How to determine the time of an implicit temporal query and use the determined time for improving search results?

Challenges with temporal queries

- *Semantic gaps*: lacking knowledge about
  1. Possibly relevant time of queries
  2. **Named entity changes over time**



Current approaches

1. Query log analysis
2. Search result analysis

Examples of the Google zeitgeist queries and associated time periods

Query	Time	Query	Time
diana car crash	1997	madrid bombing	2005
world trade center	2001	pope john paul ii	2005
osama bin laden	2001	tsunami	2005
london congestion charges	2003	germany soccer world cup	2006
john kerry	2004	torino games	2006
tsa guidelines liquids	2004	subprime crisis	2007
athens olympics games	2004	obama presidential campaign	2008

[Kanhabua et al., ECDL 2010]

Query log analysis

- Mining query logs
  - Analyze query frequencies over time for identifying the *relevant time* of queries
  - Re-rank search results of implicit temporal queries using the determined time

[Metzler et al., SIGIR 2009; Zhang et al., EMNLP 2010]

Determining time of queries

Approach I. Dating using **keywords***

Approach II. Dating using **top-k documents***
- Queries are short keywords
- Inspired by pseudo-relevance feedback

Approach III. Using **timestamp** of top-k documents
- No temporal language models are used

**Using Temporal Language Models proposed by de Jong et al.*

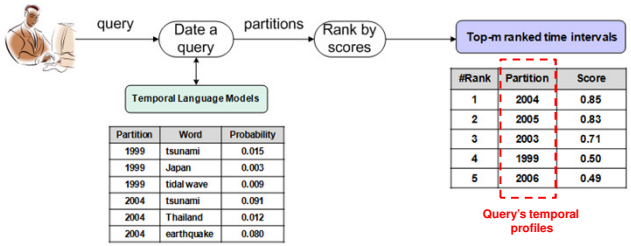
[Kanhabua et al., ECDL 2010]

Search result analysis

- Using temporal language models
  - Determine time of queries when *no time* is given explicitly
  - Re-rank search results using the determined time
- Exploiting time from search snippets
  - Extract temporal expressions (i.e., **years**) from the contents of top-k retrieved web snippets for a given query
  - Content-based language-independent approach

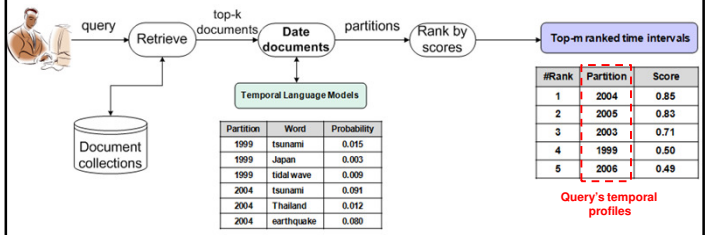
[Kanhabua et al., ECDL 2010; Campos et al., TempWeb 2012]

I. Dating using keywords



[Kanhabua et al., ECDL 2010]

II. Dating using top-k documents



Partition	Word	Probability
1999	tsunami	0.015
1999	Japan	0.003
1999	tidal wave	0.009
2004	tsunami	0.091
2004	Thailand	0.012
2004	earthquake	0.080

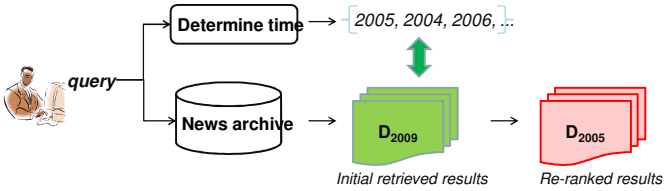
#Rank	Partition	Score
1	2004	0.85
2	2005	0.83
3	2003	0.71
4	1999	0.50
5	2006	0.49

[Kanhabua et al., ECDL 2010]

Re-ranking search results

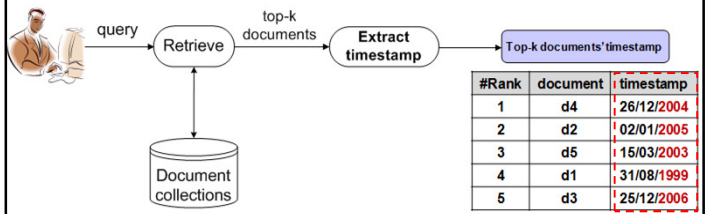
- *Intuition:* documents published **closely to the time** of queries are more relevant
  - Assign ***document priors*** based on publication dates

$$S(q,d) = (1-\alpha)\cdot S'(q_{word}, d_{word}) + \alpha \cdot S''(q_{time}, d_{time})$$



[Kanhabua et al., ECDL 2010]

III. Using timestamp of documents

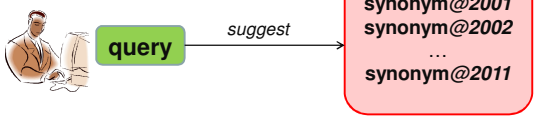


#Rank	document	timestamp
1	d4	26/12/2004
2	d2	02/01/2005
3	d5	15/03/2003
4	d1	31/08/1999
5	d3	25/12/2006

[Kanhabua et al., ECDL 2010]

Challenges of temporal search

- *Semantic gaps*: lacking knowledge about
  1. Possibly relevant time of queries
  2. **Named entity changes over time**



Named entity evolution

Problem Statements

- Queries of **named entities** (people, company, place)
  - Highly dynamic in appearance, i.e., relationships between terms changes over time
  - E.g. changes of roles, name alterations, or semantic shift

Scenario 1
Query: **"Pope Benedict XVI"** and written *before 2005*
Documents about **"Joseph Alois Ratzinger"** are relevant

Scenario 2
Query: "**Hillary R. Clinton**" and written *from 1997 to 2002*
Documents about "**New York Senator**" and "**First Lady of the United States**" are relevant

Top 10 Celebrity Name Changes	Top 10 Corporate Name Changes
1. Lisa Bonet	1. Netflix
2. Big Baby Jesus	2. Comcast
3. Whoopi Goldberg	3. Accenture
4. Mark Super Duper	4. Syfy
5. Vin Diesel	5. Royal Mail
6. Metta World Peace	6. Academi
7. Prince	7. Altria
8. Cat Stevens	8. WWE, Inc.
9. Sean Combs	9. Spike TV
10. Chad Johnson	10. ValuJet Airlines
Top 10 Dubious Name Changes	**Top 10 Geographical Name Changes**
1. Madonna	1. Belarus
2. French fries	2. Burma
3. Joseph Stalin	3. Cambodia
4. Newark Liberty International Airport	4. Bangalore, India
5. Chad Johnson	5. Chemnitz, Germany
6. Willis Tower	6. Cóbh, Ireland
7. Truth or Consequences, New Mexico	7. Ho Chi Minh City, Vietnam
8. Ed Koch Queensboro Bridge	8. Montana, Bulgaria
9. SyFy	9. Polokwane, Limpopo, South Africa
10. Sporting Kansas City	10. Saint Petersburg, Russia

QUEST Demo: http://research.idi.ntnu.no/wislab/quest/

Web Science – **Investigating the Future** of Information and Communication

Named entity evolution

Research question

- How to detect named entity changes in web documents?

Web Science – **Investigating the Future** of Information and Communication

Current approaches

1. Temporal co-occurrence
2. Temporal association rule mining
3. Temporal knowledge extraction
   - Ontology
   - Wikipedia history

Temporal co-occurrence

- Temporal co-occurrence
  - Measure the degree of relatedness of two entities at different times by comparing term contexts
  - Require a recurrent computation at querying time, which reduce efficiency and scalability

[Berberich et al., WebDB 2009]

Temporal knowledge extraction

- YAGO ontology
  - Extract named entities from the YAGO ontology
  - Track named entity evolution using the New York Times Annotated Corpus
- Wikipedia history
  - Define a time-based synonym as a term semantically related to a named entity at a particular time period
  - Extract synonyms of named entities from *anchor texts* in article links using the whole history of Wikipedia

[Mazeika et al., CIKM 2011; Kanhabua et al., JCDL 2010]

Association rule mining

- Temporal association rule mining
  - Discover semantically identical concepts (or named entities) that are used in different time
  - Two entities are semantically related if their associated events occur multiple times in a collection
  - Events are represented as sentences containing a subject, a verb, objects, and nouns

[Kaluarachchi et al., CIKM 2010]

Searching with name changes

- Extract time-based synonyms from Wikipedia
  - Synonyms are words with similar meanings
  - In this context, synonyms refer ***name variants*** (name changes, titles, or roles) of a named entity
    - E.g., "Cardinal Joseph Ratzinger" is a synonym of "Pope Benedict XVI" *before 2005*
- Two types of time-based synonyms
  1. Time-independent
  2. Time-dependent

[Kanhabua et al., JCDL 2010]

Recognize named entities

Step 1: Partition Wikipedia regarding to the time granularity $g = month$ to obtain its snapshots $\mathbb{W} = \{W_{t_1}, \ldots, W_{t_z}\}$

Step 2: For each snapshot $W_{t_k} \in \mathbb{W}$, identify named entity pages to obtain a set of named entities $E_{t_k} = \{e_1, \ldots, e_j\}$

Risk (game)

From Wikipedia, the free encyclopedia

Risk is a strategic board game, produced by Parker Brothers (now a division of Hasbro). It was invented by French film director Albert Lamorisse and originally released in 1957, as ***La Conquête du Monde*** ("The Conquest of the World"), in France.

Risk is a turn-based game for two to six players. The standard version is played on a board depicting a stylised Napoleonic-era political map of the Earth, divided into forty-two territories, which are grouped into six continents. Players control armies with which they attempt to capture territories from other players. The primary object of the game is "world domination," or "to occupy every territory on the board and in so doing, eliminate all other players."[1] Using area movement, *Risk* ignores limitations such as the vast size of the world and the logistics of long campaigns.

In the 40th Anniversary Collector's Edition the movement route between the territories of East Africa and Middle East was removed; this was later confirmed to be a manufacturing error, an error repeated in Risk II. Subsequent editions restored the missing route.[2] While the European versions of *Risk* had included the variation "*Secret Mission Risk*" for some time, the U.S. version did not have this added until 1993.[3]

[Kanhabua et al., JCDL 2010]

Initial results

- Time periods are not accurate

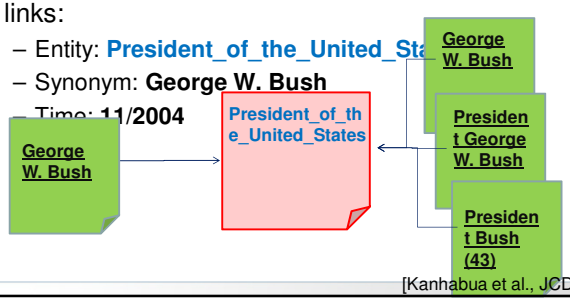
Named Entity	Synonym	Time Period
Pope Benedict XVI	**Cardinal Joseph Ratzinger**	**05/2005 - 03/2009***
	Joseph Ratzinger	05/2005 - 03/2009
	Pope Benedict XVI	05/2005 - 03/2009
Barack Obama	Barack Hussein Obama II	02/2007 - 03/2009
	Sen. Barack Obama	07/2007 - 03/2009
	Senator Barack Obama	05/2006 - 03/2009
Hillary Rodham Clinton	Hillary Clinton	08/2003 - 03/2009
	Sen. Hillary Clinton	03/2007 - 03/2009
	Senator Clinton	11/2007 - 03/2009

Note: the time of synonyms are timestamps of Wikipedia articles (8 years)

[Kanhabua et al., JCDL 2010]

Find synonyms

- Find a set of **entity-synonym relationships** at *time* t_k
- For each $e_i \epsilon E_{tk}$, extract **anchor texts** from article links:
  - Entity: **President_of_the_United_Sta**
  - Synonym: **George W. Bush**
  - Time: **11/2004**



[Kanhabua et al., JCDL 2010]

Enhancement using NYT

- Analyze **NYT Corpus** to discover accurate time
  - 20-year time span (1987-2007)
- Use the **burst detection** algorithm
  - Time periods of synonyms = burst intervals

Results from burst-detection algorithm

Synonym	Entity	Burst Weight	Time	
			Start	End
President Reagan	Ronald Reagan	5506.858	01/1987	02/1989
President Ronald	Ronald Reagan	100.401	01/1989	03/1990
President Ronald	Ronald Reagan	67.208	07/1990	02/1993
Senator Clinton	Hillary Rodham Clinton	18.214	01/2001	10/2001
Senator Clinton	Hillary Rodham Clinton	17.732	05/2002	01/2003
Senator Clinton	Hillary Rodham Clinton	172.356	06/2003	11/2004

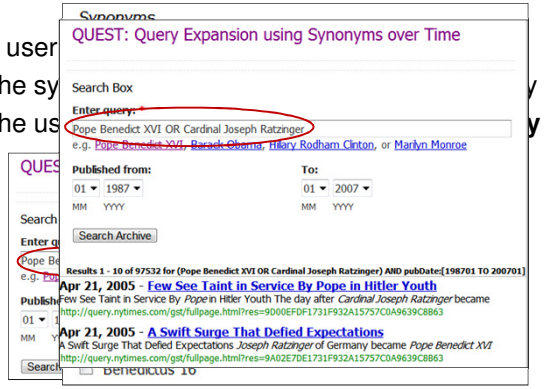
Initial results

Hillary Rodham Clinton	Hillary Clinton	08/2003 - 03/2009
	Sen. Hillary Clinton	03/2007 - 03/2009
	Senator Clinton	11/2007 - 03/2009

[Kanhabua et al., JCDL 2010]

Query expansion

1. A user
2. The sy
3. The us



QUEST Demo: http://research.idi.ntnu.no/wislab/quest/

[Kanhabua et al., ECML PKDD 2010]

Query performance prediction

Problem Statement

- Predict the **effectiveness** (e.g., MAP) that a query *will achieve* in advance of, or during retrieval
  - ***high*** MAP → "***good***"
  - ***low*** MAP → "***poor***"

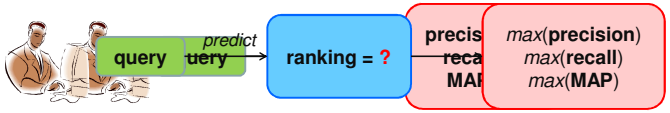
Objective

- Apply **query enhancement techniques** to improve the overall performance
  - *Query suggestion* is applied for "***poor***" queries

[Hauff et al., CIKM 2008 ;Hauff et al., ECIR 2010; Carmel et al., 2010]

Query prediction problems

1. Performance prediction
   - Predict **the retrieval effectiveness** wrt. a ranking model
2. Retrieval model prediction
   - Predict **the retrieval model** that is most suitable



Temporal query performance prediction

- **First study** of performance prediction for temporal queries
  - Propose 10 **time-based pre-retrieval** predictors
    - Both *text and time* are considered
- Experiment
  - Collection: NYT Corpus and 40 temporal queries
- Results
  - **Time-based predictors** outperform keyword-based predictors
  - **Combined predictors** outperform single predictors in most cases
- Open issue
  - Consider *time uncertainty*

[Kanhabua et al., SIGIR 2011]

Time-aware ranking prediction

- Problem statement
  - Two time aspects: **publication time** and **content time**
    - Content time = temporal expressions mentioned in documents
  - Difference in effectiveness for temporal queries when ranking using *publication time* or *content time*

Query	MAP	
	PT-Rank	*CT-Rank*
iraq 2001	**0.60**	0.40
sound of music 1960s	0.11	**0.29**
mac os x 24 march 2001	**0.79**	0.36
michael jackson 1982	0.56	**0.65**

[Kanhabua et al., SIGIR 2011]

Temporal KL-divergence

- Measure the difference between the distribution over time of top-k retrieved documents of q and the collection
  - Consider *both* time dimensions

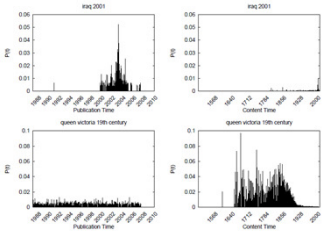


Figure 7.1: Distribution over two time dimensions of top-1000 documents for the queries iraq 2001 and queen victoria 19th century.

[Diaz et al., SIGIR 2004]

Learning to select time-aware ranking

- **First study** of the impact on retrieval effectiveness of ranking models using *two time aspects*
- Three features from analyzing *top-k* results
  - Temporal KL-divergence [Diaz et al., SIGIR 2004]
  - Content Clarity [Cronen-Townsend et al., SIGIR 2002]
  - Divergence of retrieval scores [Peng et al., ECIR 2010]

[Kanhabua et al., SIGIR 2012]

Content Clarity

- The content clarity is measured by the Kullback-Leibler (KL) divergence between the distribution of terms of retrieved documents and the background collection

[Cronen-Townsend et al., SIGIR 2002]

Divergence of ranking scores

- Measure the divergence of scores from the base ranking, e.g., a non time-aware ranking model
  - To determine the extent that a ranking model alters the scores of the initial ranking
- Features
  1. averaged scores of the base ranking
  2. averaged scores of *PT-Rank*
  3. *averaged scores of CT-Rank*
  4. *divergence* from the base ranking model

[Peng et al., ECIR 2010]

References

- [Berberich et al., WebDB 2009] Klaus Berberich, Srikanta J. Bedathur, Mauro Sozio, Gerhard Weikum: Bridging the Terminology Gap in Web Archive Search. WebDB 2009
- [Berberich et al., ECIR 2010] Klaus Berberich, Srikanta J. Bedathur, Omar Alonso, Gerhard Weikum: A Language Modeling Approach for Temporal Information Needs. ECIR 2010: 13-25
- [Campos et al., TempWeb 2012] Ricardo Campos, Gaël Dias, Alípio Jorge, Celia Nunes: Enriching temporal query understanding through date identification: How to tag implicit temporal queries? TWAW 2012: 41-48
- [Carmel et al., 2010] David Carmel, Elad Yom-Tov: Estimating the Query Difficulty for Information Retrieval Morgan & Claypool Publishers 2010
- [Cronen-Townsend et al., SIGIR 2002] Stephen Cronen-Townsend, Yun Zhou, W. Bruce Croft: Predicting query performance. SIGIR 2002: 299-306
- [Diaz et al., SIGIR 2004] Fernando Diaz, Rosie Jones: Using temporal profiles of queries for precision prediction. SIGIR 2004: 18-24
- [Hauff et al., CIKM 2008] Claudia Hauff, Vanessa Murdock, Ricardo A. Baeza-Yates: Improved query difficulty prediction for the web. CIKM 2008: 439-448
- [Hauff et al., ECIR 2010] Claudia Hauff, Leif Azzopardi, Djoerd Hiemstra, Franciska de Jong: Query Performance Prediction: Evaluation Contrasted with Effectiveness. ECIR 2010: 204-216
- [Kaluarachchi et al., CIKM 2010] Amal Chaminda Kaluarachchi, Aparna S. Varde, Srikanta J. Bedathur, Gerhard Weikum, Jing Peng, Anna Feldman: Incorporating terminology evolution for query translation in text retrieval with association rules. CIKM 2010: 1789-1792
- [Kanhabua et al., JCDL 2010] Nattiya Kanhabua, Kjetil Nørvåg: Exploiting time-based synonyms in searching document archives. JCDL 2010: 79-88
- [Kanhabua et al., ECDL 2010] Nattiya Kanhabua, Kjetil Nørvåg: Determining Time of Queries for Re-ranking Search Results. ECDL 2010: 261-272

Discussion

- Results
  - A **small number** of top-*k* documents achieves better performance
  - The larger number *k*, the more irrelevant documents are introduced into the analysis
- Open issue
  - When comparing with the optimal case there is still room for further improvements

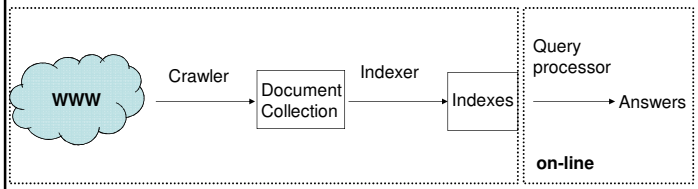
[Kanhabua et al., SIGIR 2012]

References (cont')

- [Kanhabua et al., SIGIR 2011] Nattiya Kanhabua, Kjetil Nørvåg: Time-based query performance predictors. SIGIR 2011: 1181-1182
- [Kanhabua et al., SIGIR 2012] Nattiya Kanhabua, Klaus Berberich, Kjetil Nørvåg: Learning to select a time-aware retrieval model. (To appear) SIGIR 2012
- [Li et al., CIKM 2003] Xiaoyan Li, W. Bruce Croft: Time-based language models. CIKM 2003: 469-475
- [Metzler et al., SIGIR 2009] Donald Metzler, Rosie Jones, Fuchun Peng, Ruiqiang Zhang: Improving search relevance for implicitly temporal queries. SIGIR 2009:
- [Mazeika et al., CIKM 2011] Arturas Mazeika, Tomasz Tylenda, Gerhard Weikum: Entity timelines: visual analytics and named entity evolution. CIKM 2011: 2585-2588
- [Nunes et al., ECIR 2008] Sérgio Nunes, Cristina Ribeiro, Gabriel David: Use of Temporal Expressions in Web Search. ECIR 2008: 580-584
- 700-701
- [Peng et al., ECIR 2010] Jie Peng, Craig Macdonald, Iadh Ounis: Learning to Select a Ranking Function. ECIR 2010: 114-126
- [Zhang et al., EMNLP 2010] Ruiqiang Zhang, Yuki Konda, Anlei Dong, Pranam Kolari, Yi Chang, Zhaohui Zheng: Learning Recurrent Event Queries for Web Search. EMNLP 2010: 1129-1139

Question?

Architecture of a Search Engine



- How search results change?
  1. Crawl the Web
  2. Index the content
  3. Go to line 1

Evolution of Web Search Results

(1) Short-term impacts: Caching Results

(2) Longitudinal analysis of search results

Crawling the Dynamic Web

Indexing the Dynamic Web

- How to refresh the index?
  - Batch update (re-build)
  - Re-merge
  - In-place
- Batch update
  - Shadowing
  - Simplest, the old index can keep serving at high rates

[Lester et al., IPM 2006]

Indexing the Dynamic Web

- In-place
  - Over-allocation: Leave free space at the end of each list
  - Add new entries to the free space; o.w., relocate to a new position on disk

[Lester et al., IPM 2006]

Indexing the Dynamic Web

- Re-merge
  - A "buffer" **B** of new index entries
  - Compute queries over index **I** and **B** and merge results
  - Merge **B** with **I** when size(**B**) > threshold
  - Optimizations: logarithmic and geometric merging

[Lester et al., IPM 2006]

Query Processing

- Is updating the underlying index enough?

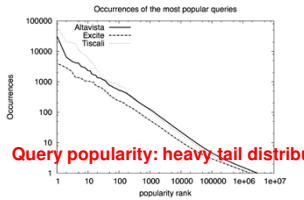
If a query is submitted for the first time, it is processed over an *up-to-date* index

But... what about the *cached* results?

Why Result Caching?

- Around 50% of a query stream is composed of repeating queries

Query Log	Queries	Distinct Queries	Date
Excite	2,475,684	1,598,908	September 16th, 1997
Tiscali	3,278,211	1,538,934	April 2002
Alta Vista	7,175,648	2,657,410	Summer of 2001

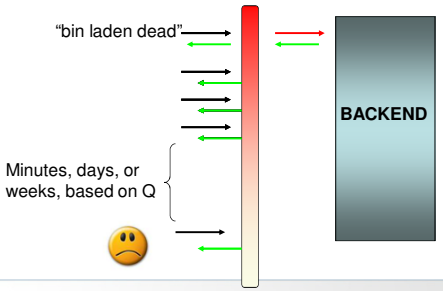


Query popularity: heavy tail distribution

[Fagni et al., TOIS 2006]

Problem: dynamicity!

- Key observations:
  - Caches can be very large (thanks to cheap hw)
  - Web changes rapidly (thanks to users)

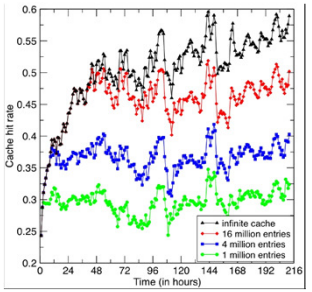


Result Caches

- Result cache: top-k urls with snippets per query
- Results caching helps to reduce
  - query response time
  - traffic volume hitting the backend

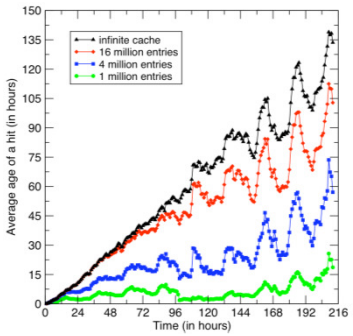
A deeper look: Capacity vs. Hits

- Query log :Yahoo! Search engine
  - Few cache nodes during 9 days
- Cache capacity
  - can be very large
  - eviction policy is less of a concern
- **Hit rate:** fraction of queries that are hits
- Higher capacity → more hits!



[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

Capacity vs. age

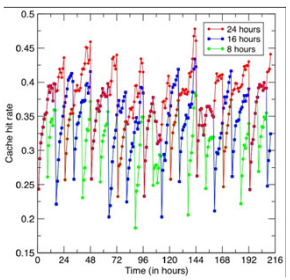


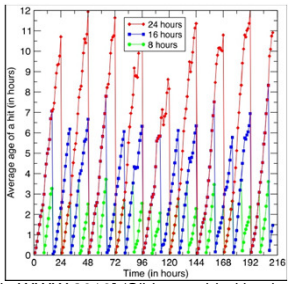
- Hit age: time since the last update
- Higher capacity → higher age!

[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

Decoupled approaches: Flushing

- Solution: flush periodically
  - Coincides with new index
  - Bounds average age
  - Impacts hit rate negatively





[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

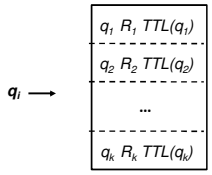
Strategies for cache freshness

- **Decoupled approaches:** cache does not know what changed
- **Coupled approaches**: cache uses clues from the index to predict what changed

[Cambazoglu et al., WWW 2010]

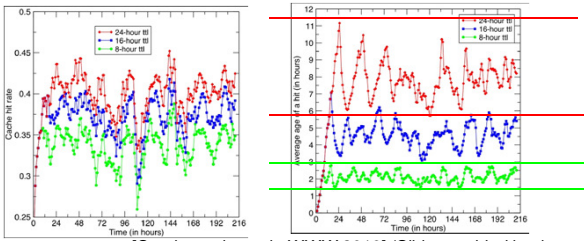
Decoupled approaches:TTL

- Time-to-live (TTL)
  - Assign a fixed TTL value to each cached result
  - **Pros:** Practical, almost no implementation cost
  - **Cons:** Blind strategy, may be sub-optimal



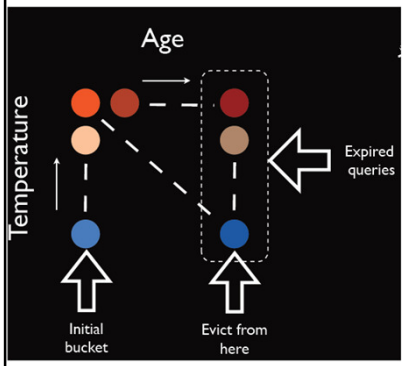
Decoupled approaches:TTL

- TTL per query
  - Stale results → Acceptable: search engines do not have a perfect view!
  - Stable hit rate; average age is still bounded!



[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

Intelligent refreshing



- A new query goes to young and cold bucket!
- Fixed "query interval" for shifting age
- Lazy update of temperature
- Refresh hot and old first!

[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

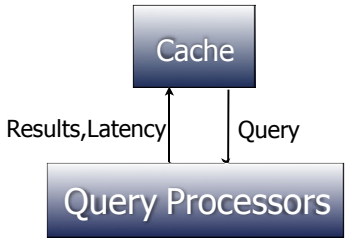
Intelligent Refreshing

- Enhancement to the TTL mechanism
- Updates entries
  - Re-execute queries
  - Uses idle cycles
- Ideally
  - update: low activity
  - use: high activity
- How to select queries for refreshing?

[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

Refresh-rate adjustment

- Idle cycles?
- Critical mechanism
  - Prevent overloads
- Feedback from the query processors
  - Latency to process query
- Track recent latency
  - Adjust rate accordingly



[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

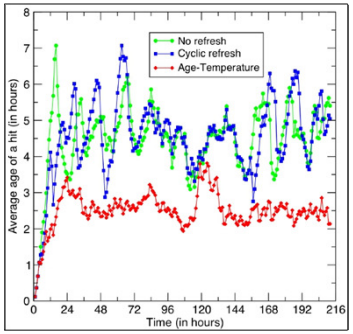
Design summary

- Large capacity: millions of entries
- TTL: bounds the age of entries
- Refreshes: updates cache entries
- Refresh rate adjustment: latency feedback

[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

Cache evaluation

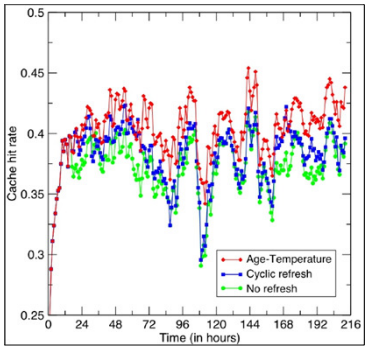
- Simulation
  - Yahoo! query log
- Baseline policies
  - No refresh
  - Cyclic refresh



[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

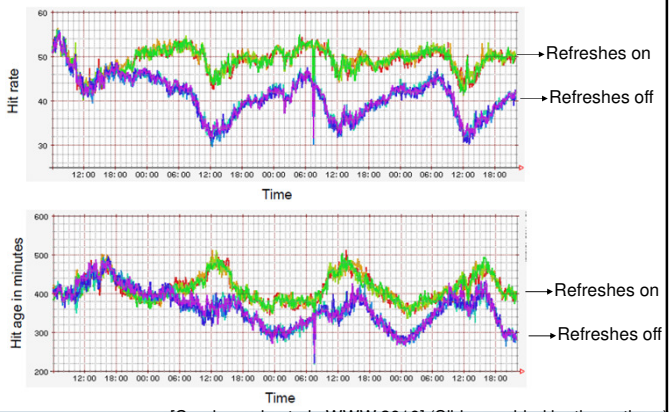
Cache evaluation

- Simulation
  - Yahoo! query log
- Baseline policies
  - No refresh
  - Cyclic refresh



[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

Performance in production



[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

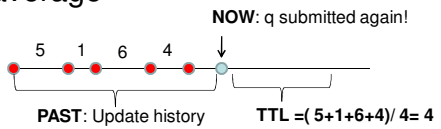
Degradation

- Under high load
  - Processors degrade results
  - Shorter life
- TTL mechanism
  - Prioritize for refreshing
  - AdjustsTTL when degraded
- Refreshes
  - Replace with better results

[Cambazoglu et al., WWW 2010] (Slide provided by the authors)

Adaptive I: Average TTL

- Observe past update frequency of for top-k results of a query and compute average



- Simple, but
  - Needs history
  - May not capture bursty update periods

[Alici et al., ECIR 2012] (Slide provided by the authors)

Adaptive TTL

- Up to now we considered fixed TTL values
- Are all queries equal?
  - "quantum physics"
  - "barcelona FC"
  - "ecir"

One size does not fit all!!!

- Another promising direction is assigning adaptive TTL values

[Alici et al., ECIR 2012] (Slide provided by the authors)

Adaptive II: Incremental TTL

- Adjust the new TTL value based on the current TTL value
- Each time a cached result R_{cached} with an expired TTL_{curr} is requested:

compute R_{new}

if $R_{new} \neq R_{cached}$ // **STALE**

$TTL_{new} \leftarrow TTL_{min}$ // catch bursty updates!

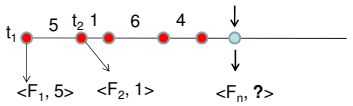
else

$TTL_{new} \leftarrow TTL_{curr} + F(TTL_{curr})$ // F(.): linear, poly., exp.

[Alici et al., ECIR 2012] (Slide provided by the authors)

Adaptive III: Machine-learned

- Build an ML model
  - Query and result-specific feature set F
  - Assume a query result changes at time point t_i
    - Create a training instance with features F_i
    - Set target feature (TTL_{new}) as the time period from t_i to the time point t_j where query result changes again: t_j-t_i



[Alici et al., ECIR 2012] (Slide provided by the authors)

Analysis of Web Search Results

- Setup:
  - 4,500 queries sampled from AOL log
  - Submitted to Yahoo! API daily from Nov 2010 to April 2011 (120 days)
  - For days d_i and d_{i+1}

    If $R_i \neq R_{i+1}$ (consider top-10 urls and their order), we consider the result as **updated**!

[Alici et al., ECIR 2012] (Slide provided by the authors)

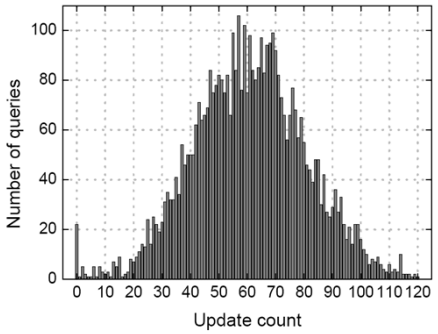
Features for ML

Table 2. Features used by the machine learning model

Feature	Description
`QueryTermCount`	Number of terms in the query
`QueryFreqeuency`	Number of times query appears in the log
`ResultCount`	Number of query results
`ReplacedDocsInResult`	Number of new query results
`RerankedDocsInResult`	Number of query results whose ranks have changed
`Web2ResultCount`	Number of query results from Web 2.0 sites
`NewsResultCount`	Number of query results from news sites

[Alici et al., ECIR 2012] (Slide provided by the authors)

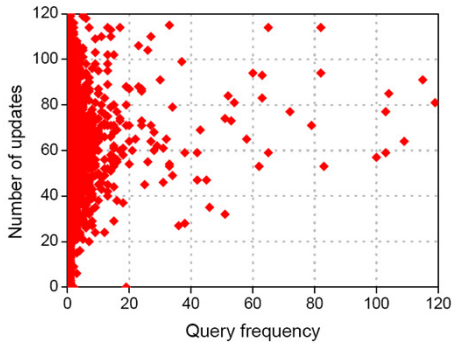
Distribution of result updates



- On average, query results are updated in every 2 days!

[Alici et al., ECIR 2012] (Slide provided by the authors)

Effect of query frequency



- More frequent → results updated more frequently
- Less frequent → scattered

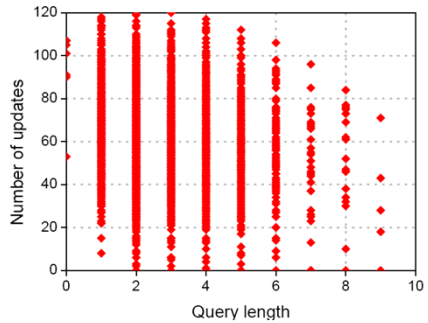
[Alici et al., ECIR 2012] (Slide provided by the authors)

Simulation

- Assume all 4,500 queries are submitted daily for 120 days
  - On day-0, all results are cached
  - On the following days, whenever the TTL expires, the new result is computed and replaces old ones (i.e., result for that day from Yahoo! API)

[Alici et al., ECIR 2012] (Slide provided by the authors)

Effect of query length



- No correlation between query length and update frequency

[Alici et al., ECIR 2012] (Slide provided by the authors)

Simulation setup

- Evaluation metrics [Blanco 2010]
  - At the end of each day, we compute:

$$\textbf{False Positive Ratio} = \frac{\text{Redundant query executions}}{\text{Number of unique queries}}$$

$$\textbf{Stale Traffic Ratio} = \frac{\text{Stale results returned}}{\text{Number of query occurrences}}$$

[Alici et al., ECIR 2012] (Slide provided by the authors)

Simulation setup

- While computing ST ratio for a given query at day *i*:

Strict policy:

– if R_{cached} is not strictly equal to $R_{day\text{-}i}$
staleCount++

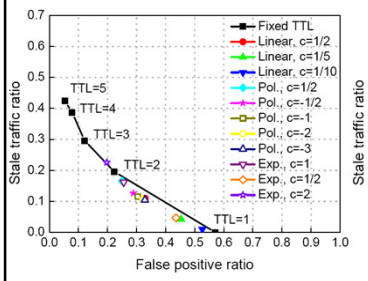
Relaxed policy:

– if R_{cached} is not strictly equal to $R_{day\text{-}i}$
staleness += 1 – JaccardSim(R_{cached}, $R_{day\text{-}i}$)

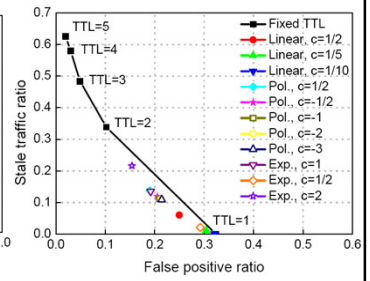
[Alici et al., ECIR 2012] (Slide provided by the authors)

Performance: Incremental TTL

Less-often updated queries

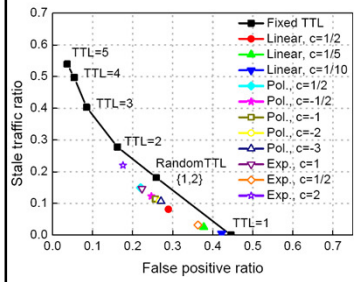


More-often updated queries

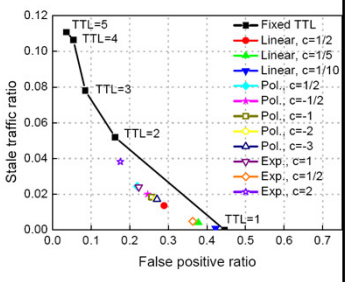


[Alici et al., ECIR 2012] (Slide provided by the authors)

Performance: Incremental TTL



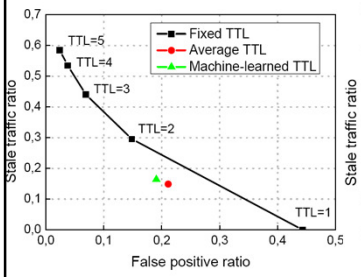
Strict policy



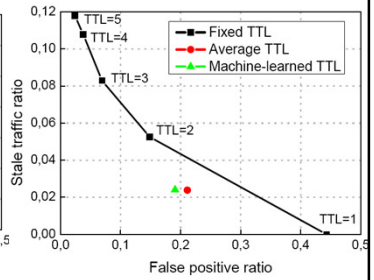
Relaxed policy

[Alici et al., ECIR 2012] (Slide provided by the authors)

Performance: Average and Machine Learned TTL



Strict policy



Relaxed policy

[Alici et al., ECIR 2012] (Slide provided by the authors)

Coupled approaches: CIP

- Cache invalidation policy (CIP)
- Key idea: Compare all added/updated/deleted documents to cached queries
  - Incremental index update → updates reflected on-line!

all queries in cache(s) → **CIP module** ← all changes in the backend index

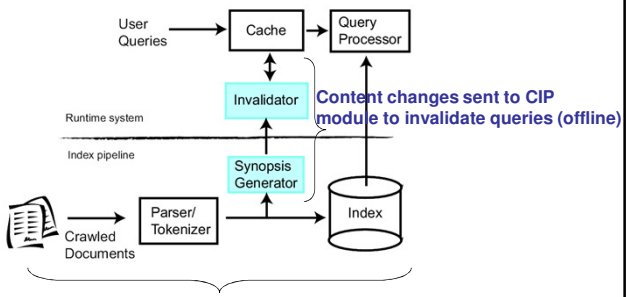
[Blanco et al., SIGIR 2010]

Synopses

- Synopsis: A vector of document's top-scoring TF-IDF terms
  - η: length of synopsis
  - δ: modification threshold → consider a document d as updated only if $diff(d, d_{old}) > \delta$

[Blanco et al., SIGIR 2010]

Coupled approaches: CIP



Incremental index update: updates reflected on-line!

[Blanco et al., SIGIR 2010]

Invalidation Approaches

- Invalidation logic (for added docs)

Basic: If the synopis matches the query (i.e., involve all query terms)

Scored: Compute Sim(synopsis, query) to the score of k-th result

Example

[Blanco et al., SIGIR 2010]

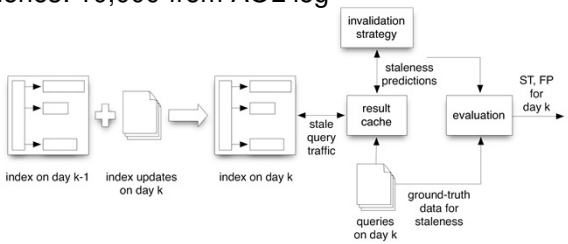
Invalidation Approaches

- Invalidation logic (for deleted docs):
  - Invalidate all query results including a deleted document
- Update is a deletion followed by an addition
- Further apply TTL to guarantee an age-bound
  - Scored + TTL

[Blanco et al., SIGIR 2010]

Simulation Setup

- Data: English wikipedia dump
  - snapshot at Jan 1, 2006 ≈ 1 million pages
  - All add/deletes/updates for following 30 days
- Queries: 10,000 from AOL log



[Blanco et al., SIGIR 2010; Alici et al. SIGIR 2011]

Evaluation

- Really hard to evaluate if you are not sitting at the production department in a real search engine company!

bing YAHOO! Google Яндекс Найдётся всё

Simulation setup

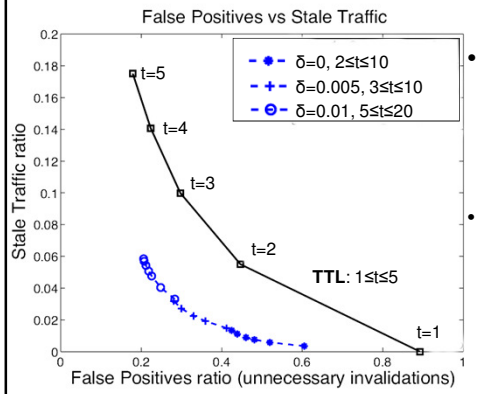
- Evaluation metric
  - The query result is updated if two top-10 lists are not *exactly* the same

$$\textbf{False Positive Ratio} = \frac{\text{Redundant query executions}}{\text{Number of unique queries}}$$

$$\textbf{Stale Traffic Ratio} = \frac{\text{Stale results returned}}{\text{Number of query occurrences}}$$

[Blanco et al., SIGIR 2010]

Performance



- Setup
  - Score + TTL
  - Full synopsis
- Smaller δ or TTL:
  - Lower ST, higher FP

[Blanco et al., SIGIR 2010]

Timestamp-based Invalidation

- The value of the TS on an item shows the *last time* the item was updated
- TIF has two components:
  - Offline (indexing time) : Decide on term and document timestamps
  - Online (query time): Decide on the staleness of the query result

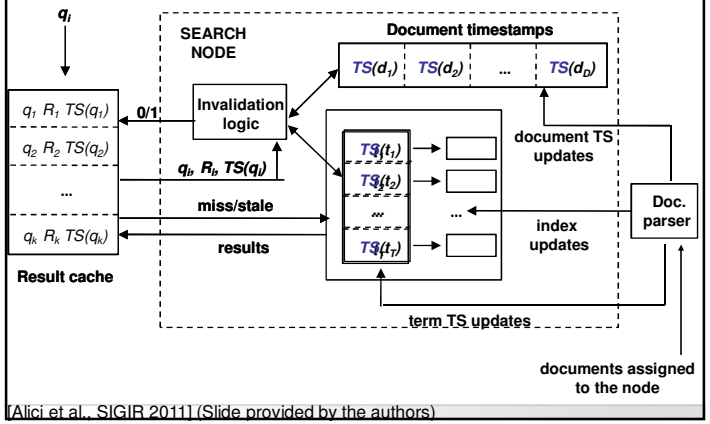
[Alici et al., SIGIR 2011] (Slide provided by the authors)

TIF: Timestamp-based Invalidation Framewrok

- Devise a new invalidation mechanism
  - better than TTL and close to CIP in detecting stale results
  - better than CIP and close to TTL in efficiency and practicality

[Alici et al., SIGIR 2011] (Slide provided by the authors)

TIF Architecture



[Alici et al., SIGIR 2011] (Slide provided by the authors)

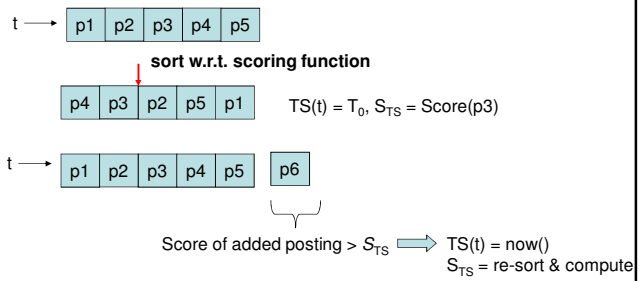
TS Update Policies: Documents

- For a **newly added** document *d*
  - TS(*d*) = now()
- For a **deleted** document *d*
  - TS(*d*) = infinite
- For an **updated** document *d*
  - if diff(d_{new}, d_{old}) > *L*
    TS(*d*) = now()
  - diff(d_i, d_j): |length(d_i) – length(d_j)|

[Alici et al., SIGIR 2011] (Slide provided by the authors)

TS Update Policies: Terms

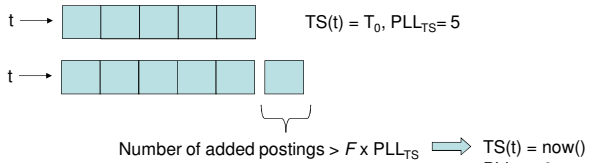
- Score based update



[Alici et al., SIGIR 2011] (Slide provided by the authors)

TS Update Policies: Terms

- Frequency based update



[Alici et al., SIGIR 2011] (Slide provided by the authors)

Result Invalidation Policy

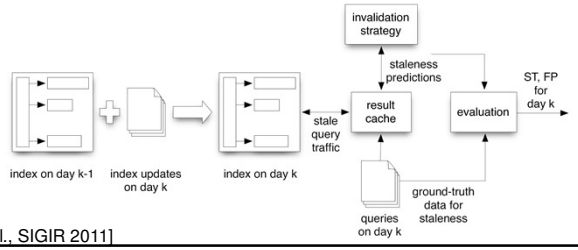
- A search node decides a result stale if:
  - C1: ∃*d* ϵ *R*, s.t. TS(*d*) > TS(*q*)
    (*d* is deleted or revised after the generation of query result)

  or,
  - C2: ∀*t* ϵ q, s.t. TS(*t*) > TS(*q*)
    (all query terms appeared in new documents after the generation of query result)
- Also apply TTL to avoid stale accumulation

[Alici et al., SIGIR 2011] (Slide provided by the authors)

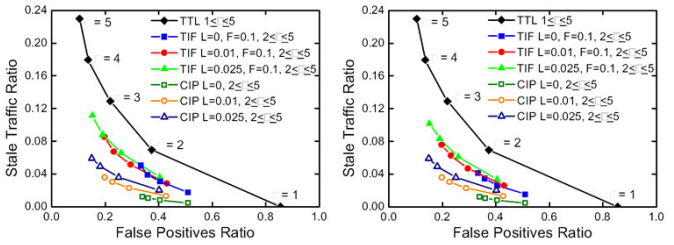
Simulation Setup

- Data: English wikipedia dump
  - snapshot at Jan 1, 2006 ≈ 1 million pages
  - All add/deletes/updates for following 30 days
- Queries: 10,000 from AOL log



[Alici et al., SIGIR 2011]

Performance: all queries



Frequency-based term TS update | Score-based term TS update

[Alici et al., SIGIR 2011] (Slide provided by the authors)

Simulation setup

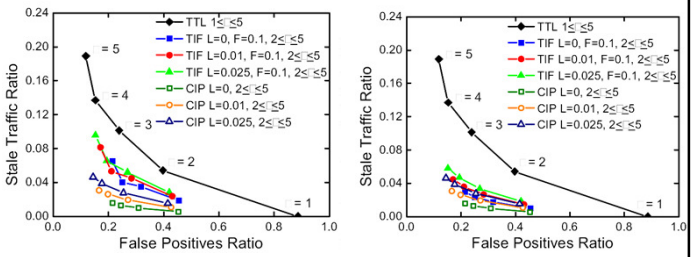
- Evaluation metrics [Blanco 2010]
  - The query result is updated if two top-10 lists are not *exactly* the same

False Positive Ratio = $\frac{\text{Redundant query executions}}{\text{Number of unique queries}}$

Stale Traffic Ratio = $\frac{\text{Stale results returned}}{\text{Number of query occurrences}}$

[Alici et al., SIGIR 2011] (Slide provided by the authors)

Performance: single-term queries



Frequency-based term TS update | Score-based term TS update

[Alici et al., SIGIR 2011] (Slide provided by the authors)

Invalidation Cost

	TIF	CIP
Data transfer	Send *<q, R, TS(q)>* to the search nodes	Send *all <q, R>* to CIP Send *all* docs to CIP
Invalidation operations	Compare TS values	Traverse the query index for every document

[Alici et al., SIGIR 2011] (Slide provided by the authors)

Grand Summary

- Fixed TTL
  - With refreshing
- Adaptive TTL
- TIF
- CIP

Decoupled	Hit rate Hit age	Yahoo!
Decoupled	ST Ratio FP Ratio	Web sample AOL
Coupled	ST Ratio FP Ratio	Wikipedia AOL
Coupled	ST Ratio FP Ratio	Wikipedia AOL

ST and FP ratio decreases

Invalidation cost decreases

Performance: TIF

- A simple yet effective invalidation approach
- Predicting stale queries
  - Better than TTL, close to CIP
- Efficiency and practicality
  - Straightforward in a distributed system

Open Research Directions

Investigate:

- User satisfaction vs. freshness
- Complex ranking functions
- Alternative index update strategies
- Combinations
  - Adaptive TTL + TIF or CIP
  - Adaptive TTL + refresh strategy

Web Science – **Investigating the Future** of Information and Communication

Evolution of Web Search Results

- Earlier works consider the changes in the content of the
  - Web
  - queries
- How do real life search engines react this dynamicity?
- Compare results from Yahoo! API
  - for 630K real life queries (from AOL log)
  - obtained in 2007 and 2010

[Altingovde et al., SIGIR 2011] (Slide provided by the authors)

Web Science – **Investigating the Future** of Information and Communication

We Seek to Answer:

- How is the growth in Web reflected to top-ranked query results?
- Do the query results totally change within time?
- Are results located deeper in sites?
- Is there any change in result title and snippet properties?

[Altingovde et al., SIGIR 2011] (Slide provided by the authors)

Web Science – **Investigating the Future** of Information and Communication

What is Novel?

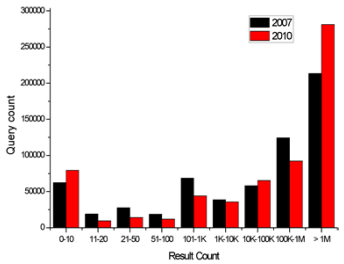
- Queries are real, not synthetic
- Query set is large
- Results from a search engine at two very distant points in time
- Focus on the properties of results, but not the underlying content

[Altingovde et al., SIGIR 2011] (Slide provided by the authors)

Web Science – **Investigating the Future** of Information and Communication

Avg no. of results

- Almost tripled (from 16M to 52M), but not uniformly



[Altingovde et al., SIGIR 2011] (Slide provided by the authors)

No. of unique URLs

- 20% of the URLs returned at the highest rank in 2010 were at the same position in 2007!

	2007	2010	Overlap (% w.r.t 2010)
Top-1	475,860	437,483	87,248 (19.9%)
Top-10	4,377,299	4,456,026	476,649 (10.7%)
Top-20	8,330,692	8,737,776	836,125 (9.6%)
Top-100	34,576,357	39,437,931	3,384,122 (8.6%)

[Altingovde et al., SIGIR 2011] (Slide provided by the authors)

Research Directions

- How are the results are diversified at these two different time points?
  - Can we deduce these from snippets?
- How does the level of bias changes in query results?

[Altingovde et al., SIGIR 2011] (Slide provided by the authors)

No. of unique domains

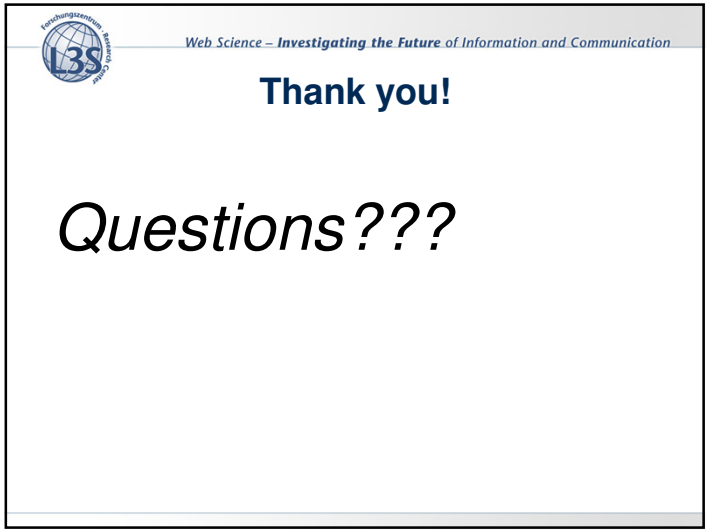
- The increase in unique domain names in 2010 is more emphasized in comparison to the increase in the number of unique URLs (diversity? coverage?)
- Even higher overlap for top-1 domains

	2007	2010	Overlap (% w.r.t 2010)
Top-1	230,464	242,859	90,040 (37.1%)
Top-10	1,065,881	1,362,538	373,811 (27.4%)
Top-20	1,678,452	2,249,991	599,280 (26.6%)
Top-100	4,462,468	6,599,437	1,705,899 (25.8%)

[Altingovde et al., SIGIR 2011] (Slide provided by the authors)

References

- **[Alici et al., SIGIR 2011]** Sadiye Alici, Ismail Sengör Altingövde, Rifat Ozcan, Berkant Barla Cambazoglu, Özgür Ulusoy: Timestamp-based result cache invalidation for web search engines. SIGIR 2011: 973-982
- **[Alici et al., ECIR 2012]** Sadiye Alici, Ismail Sengör Altingövde, Rifat Ozcan, Berkant Barla Cambazoglu, Özgür Ulusoy: Adaptive Time-to-Live Strategies for Query Result Caching in Web Search Engines. ECIR 2012: 401-412
- **[Altingovde et al., SIGIR 2011]** Ismail Sengör Altingövde, Rifat Ozcan, Özgür Ulusoy: Evolution of web search results within years. SIGIR 2011: 1237-1238
- **[Blanco et al., SIGIR 2010]** Roi Blanco, Edward Bortnikov, Flavio Junqueira, Ronny Lempel, Luca Telloli, Hugo Zaragoza: Caching search engine results over incremental indices. SIGIR 2010: 82-89
- **[Cambazoglu et al., WWW 2010]** Berkant Barla Cambazoglu, Flavio Paiva Junqueira, Vassilis Plachouras, Scott A. Banachowski, Baoqiu Cui, Swee Lim, Bill Bridge: A refreshing perspective of search engine caching. WWW 2010: 181-190
- **[Fagni et al., TOIS 2006]** Tiziano Fagni, Raffaele Perego, Fabrizio Silvestri, Salvatore Orlando: Boosting the performance of Web search engines: Caching and prefetching query results by exploiting historical usage data. ACM Trans. Inf. Syst. 24(1): 51-78 (2006)
- **[Lester et al., IPM 2006]** Nicholas Lester, Justin Zobel, Hugh E. Williams: Efficient online index maintenance for contiguous inverted lists. Inf. Process. Manage. 42(4): 916-933 (2006)


Web Science – Investigating the Future of Information and Communication
Thank you!
Questions???