Keyword Search over Dynamic Categorized Information

Manish Bhide, Venkatesan T. Chakaravarthy
(IBM India Reasearch Lab., New Delhi)

Krithi Ramamritham (IIT Bombay, Mumbai, India)

Prasan Roy (Aster Data Systems, CA, USA)

presented by

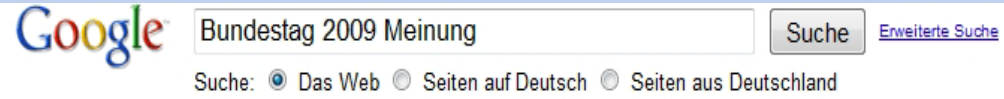
Michael Kaczmarczyk

Outline

- Motivation
- Ranking categories
- Maintaining data
- Answer Queries
- Evaluation
- Conclusion

Outline

- **Motivation**
- Ranking categories
- Maintaining data
- Answer Queries
- Evaluation
- Conclusion


Google

Bundestag 2009 Meinung

Suche | Erweiterte Suche

Suche: Das Web | Seiten auf Deutsch | Seiten aus Deutschland

Web Ergebnisse **1** - **9** von ungefähr **9.270.000** für **Bundestag 2009 Meinung**. (**0,22** Sekunden)

Bundestagswahl 2009
Informationen rund um die Wahl zum 17. Deutschen **Bundestag** am 27. September **2009**, bei der etwa 62,2 Millionen Deutsche wahlberechtigt sind.
www.**bundestag**.de/btg_wahl/index.html - Im Cache - Ähnlich

Deutscher **Bundestag**: Sitzungswochen im Deutschen **Bundestag 2009**
Deutscher **Bundestag** - **Bundestag**, Sitzungswochen, **2009**.
www.**bundestag**.de/**bundestag**/plenum/sitzungskalender/bt**2009**.html

Fragen und Antworten nach der **Bundestagswahl 2009**
September **2009** veröffentlicht. Sie ist danach für die Öffentlichkeit unter **...**
www.**bundestag**.de/btg_wahl/faq_wahl.html

Weitere Ergebnisse von bundestag.de »

Meinung - Bundestagswahl 2009 - sueddeutsche.de
Bundestagswahl 2009 Bundestagswahl 2009,Angela Merkel (CDU) gegen Frank-Walter Steinmeier (SPD): Das Duell ist Geschichte, die **Bundestagswahl 2009 ...**
www.sueddeutsche.de/politik/172/455845/uebersicht/17/ - Ähnlich

Bundestagswahl 2009 - sueddeutsche.de
Angela Merkel (CDU) gegen Frank-Walter Steinmeier (SPD): Das Duell ist Geschichte, die **Bundestagswahl 2009** entschieden. sueddeutsche.de informiert sie - mit **...**
www.sueddeutsche.de/politik/172/455845/uebersicht/ - Ähnlich

Weitere Ergebnisse anzeigen von www.sueddeutsche.de

Bundestagswahl 2009 – Wikipedia
11.1 Endgültiges Gesamtergebnis der **Bundestagswahl 2009 .....** Juli **2009**; ↑ Paul Wursch: Macht der **Meinungsumfragen** – Die Droge Demoskopie. **...**
Parteien - Personalentscheidungen der ... - Koalitionsaussagen
de.wikipedia.org/wiki/**Bundestagswahl_2009** - Im Cache - Ähnlich

Die **Bundestagswahl 2009** | tagesschau.de
Analysen, Porträts, Interviews und Hintergründe zum Thema.
www.tagesschau.de/**bundestagswahl**/ - Im Cache - Ähnlich

Bundestagswahl 2009

Documents found in categories:

teachers (1879)

students (1403)

politicians (985)

parties (560)

reporters (356)

Categorized Search

- Search method
- Returns categories, containing data items
- Items containing terms, entry date
- Categories with predicate for data items

Ranking categories

Scoring function:

$$Score\ (c, Q) = G(F(c, t_1), F(c, t_2), \ldots, F(c, t_l))$$

- $Q = \{t_1, t_2, \ldots, t_l\}$
- F computes score of c with respect to t
- G combines F for each t ϵ Q

Ranking categories

Problem:

- High data arrival
- Changing data item set D_s and data-set $M_s(c)$ for every new item
- Data must be up-to-date

Updating data

- Update: bottleneck for data input
- Assume 25ms for categorization
  1000 categories
  ->25 seconds per item

-> falling behind

Updating data

- CS* approach
  - uses a selective update strategy
  - identifies important categories
  - uses data items with high benefit

Updating data

- CS* approach
  - Maintaining statistics to increase accuracy (e.g. workload on terms)
  - Scoring of results by standard technique

Outline

- Motivation
- **Ranking categories**
- Maintaining data
- Answer Queries
- Evaluation
- Conclusion

Scoring function:

of term t in data set

$$tf_s(c,t) = \frac{\sum_{d:d\in M_s(c)} f(d,t)}{\sum_{t'\in T}\left(\sum_{d:d\in M_s(c)} f(d,t')\right)}$$

Normalized by # of terms

all categories

$$idf_s(t) = 1 + \log\left(\frac{|C|}{|C'|}\right)$$

categories with t

- Scales down terms occurring more frequent across categories

Ranking categories

Scoring function:

$$Score_s(c,Q) = \sum_{t_i \in Q} (tf_s(c,t_i) * idf_s(t_i))$$

- $Q = \{t_1, t_2, \ldots, t_l\}$

What about sampling?

- Accuracy given by Chernoff Bounds
- High accuracy leads to vast sampling numbers
- E.g. accuracy of 99%:

  => leads to 46 millions! samples needed

Outline

- Motivation
- Ranking categories
- **Maintaining data**
- Answer Queries
- Evaluation
- Conclusion

Meta-Data Refresher

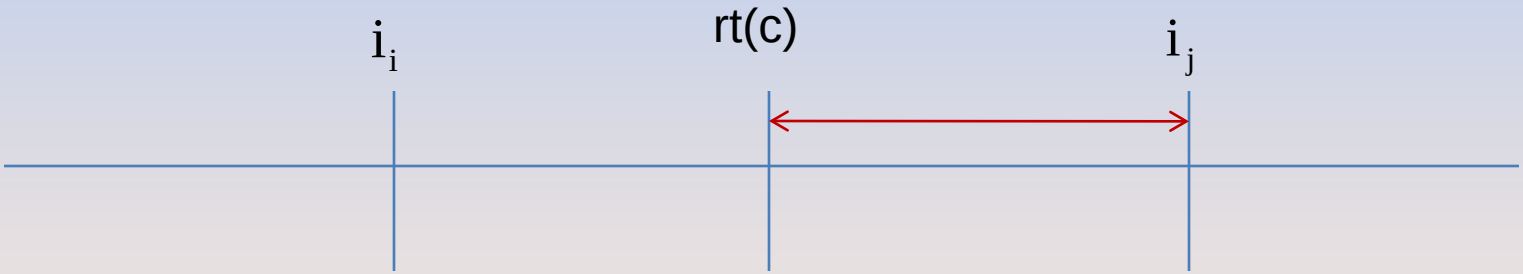
- Determines set of important categories (N)
  - by query workload $\mathcal{W}$ for t
- Keeping track of queries

# of t in W

$$\text{Importance}(c) = \sum_{t \in W \wedge c \in CandidateSet(t)} weight(t)$$

Meta-Data Refresher

- B items used for updating category
- Choosing of items using rt(c) and $[i_i, i_j] \in R$



$$Benefit([i_1, i_2]) = \sum_{c \in IC} \text{Importance(c) x Benefit}([i_1, i_2], c)$$

$rt(c)$: refresh time of c

Meta-Data Refresher

- Set of ranges $\binom{s*}{2}$
- Reduce amount of ranges to $\binom{N}{2}$
  - order categories by last refresh time
  - find nice ranges $[rt(c_i), rt(c_j)]$

    -> width(R) ≤ B and max Benefit(R)

Meta-Data Refresher

Justification:

items since last refresh time

$$\sum_{c \in \mathrm{IC}} (s* - rt(c))$$

- Much larger
- More computing power needed
- More time needed

Meta-Data Refresher

Justification:

$$E[k,b] = \max\left\{ \begin{array}{c} E[k-1,b] \\ \max\limits_{1 \le j < k} \left\{ Benefit(NR_{jk}) + E[j, b - Width(NR_{jk})] \right\} \end{array} \right\}$$

- Matrix E has N*B entries
- Runtime reduced to $O(N^2 * B)$ instead of $O((s^*)^2 * B)$ with $s^* >> N$

Meta-Data Refresher

- N and B chosen by knowing
  - γ, units of time needed for single category
  - p, processing power
  - α, data items/time unit

$$\frac{B * N * \gamma}{p} = \frac{1}{\alpha} \Rightarrow N = \frac{p}{\alpha * B * \gamma}$$

- B computed by considering staleness of N

Outline

- Motivation
- Ranking categories
- Maintaining data
- **Answer Queries**
- Evaluation
- Conclusion

Query Answering Module

- Given query $Q = \{t_1, t_2, \ldots, t_l\}$

- Goal: find top-K categories using

$$Score_{s*}^{est}(c, Q) = \sum_{t_i \in Q} (tf_{s*}^{est}(c, t_i) * idf_{s*}^{est}(t_i))$$

changing with adding item

no significant change in time

Answering Single Keywords

- Given keyword t
- Score of a category c

$$Score_{s*}^{est} = tf_{s*}^{est}(c,t) * idf_{s*}^{est}(t)$$

keeps changing

- -> hard to maintain $tf_{s*}^{est}(\cdot,t)$

Answering Single Keywords

- Given a Query $Q = \{t_1, t_2, \ldots, t_l\}$
- Needing $tf_{s*}(c, t_i)$, but $tf_{rt(c)}(c, t_i)$ available

$$tf_{s*}^{est}(c, t_i) = tf_{rt(c)}(c, t_i) + \Delta(c, t_i) \text{x} (s* - rt(c))$$

known

Estimated linear function

Answering Single Keywords

- Closer look at the tf-value

$$tf_{s*}^{est}(c,t) = tf_{rt(c)}(c,t) + \Delta(c,t)\ x\ (s* - rt(c))$$

$$= \boxed{tf_{rt(c)}(c,t) - \Delta(c,t)\ x\ rt(c)} + \boxed{\Delta(c,t)\ x\ s*}$$

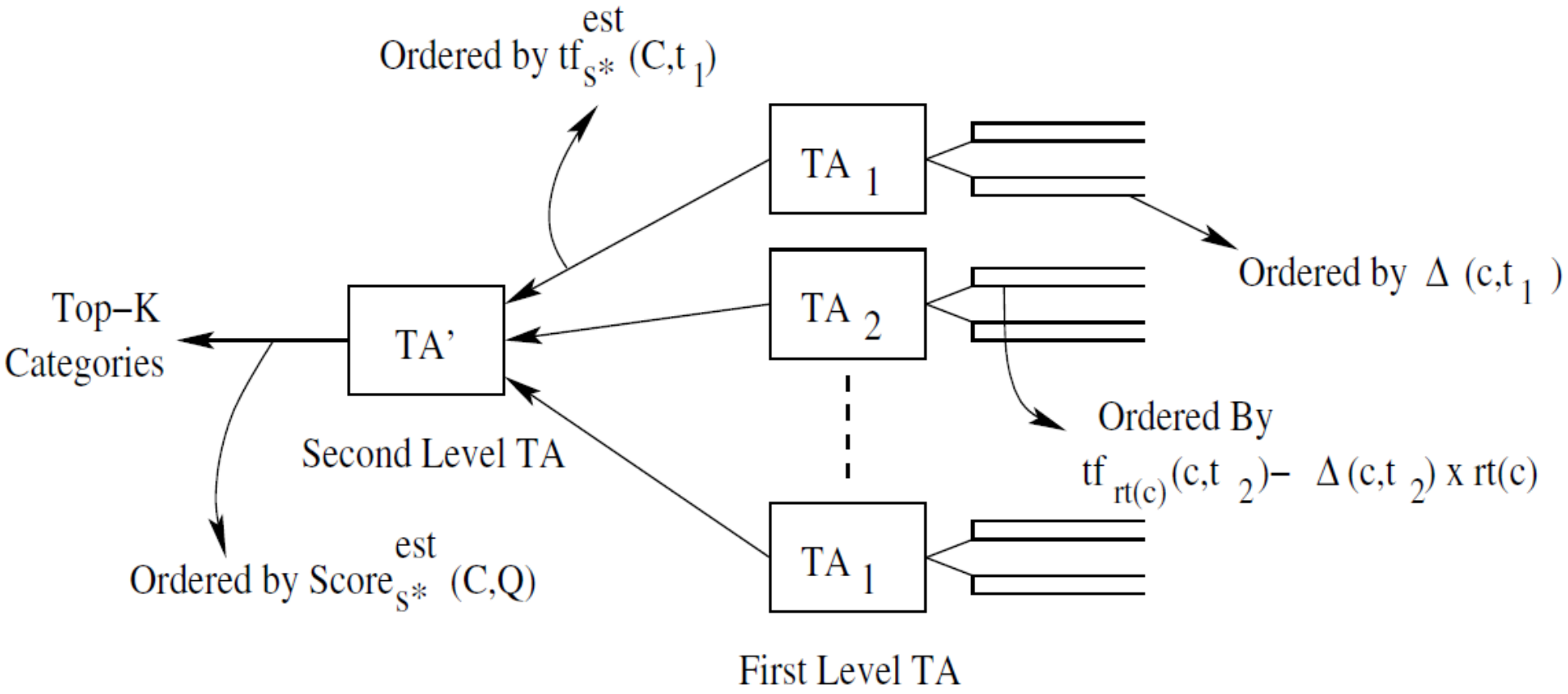
changing after refreshing c

common for all c

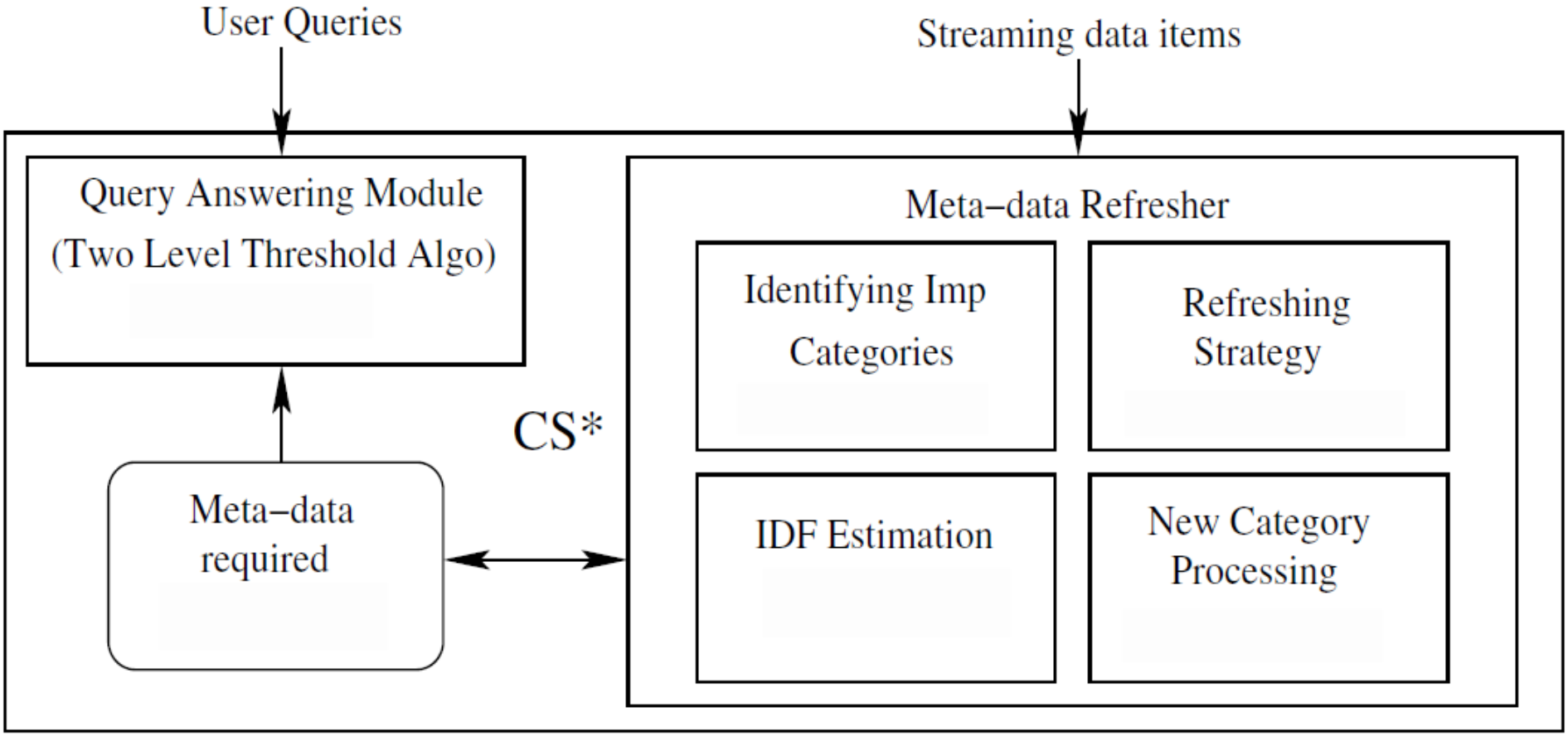
Answering Single Keywords

- Using mapping of term to category
- For a term two lists for both components
  $(tf_{rt(c)}(c,t) - \Delta(c,t)\, x\, rt(c),\ \Delta(c,t)\, x\, s^*)$
- Merged to compute top-K categories

Answering Multiple Keywords



Overview of Categorized Search



Outline

- Motivation
- Ranking categories
- Maintaining data
- Answer Queries
- **Evaluation**
- Conclusion

Evaluation

Experiment:

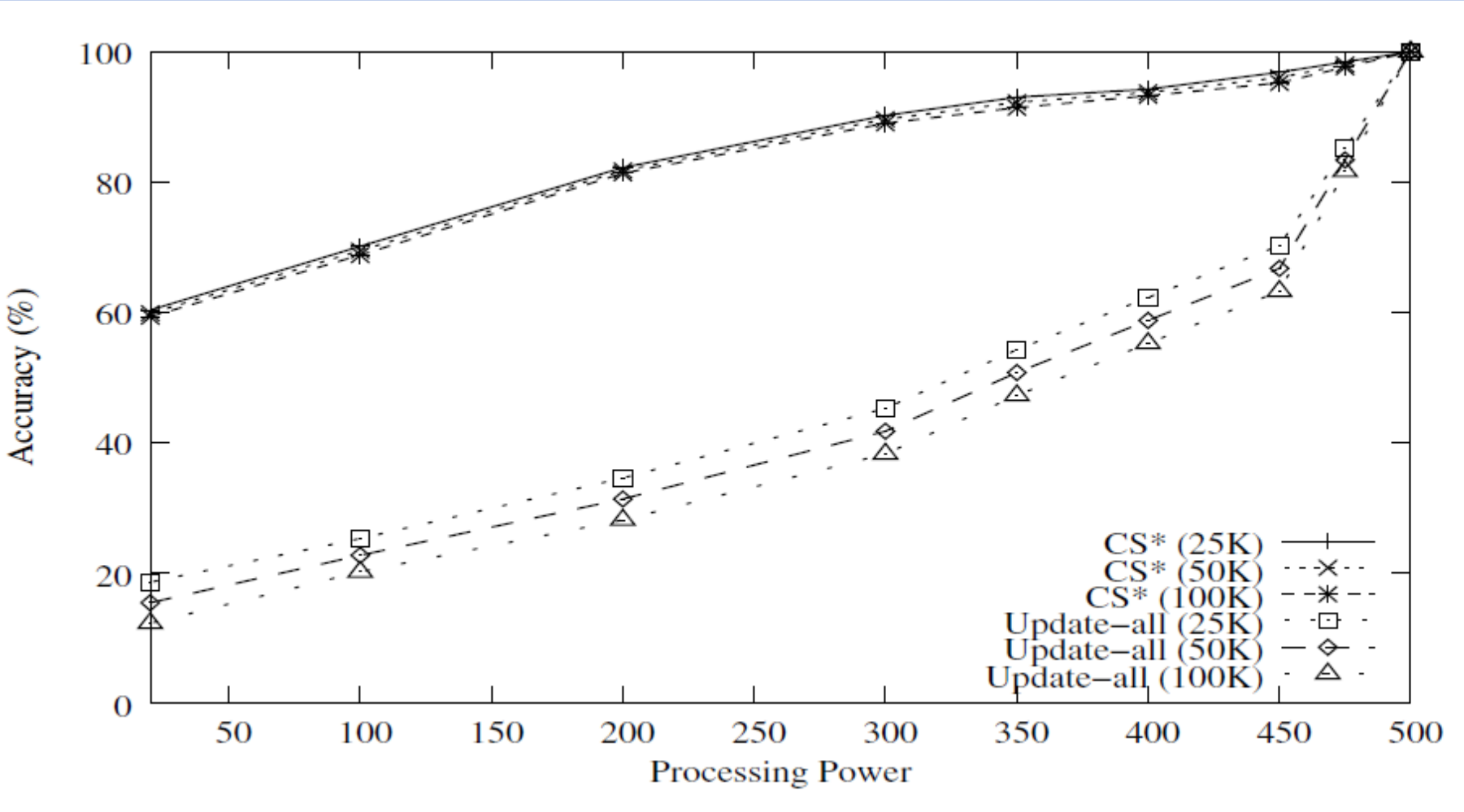
- data from www.citeulike.org since 2007
- 100,000 articles crawled
- 5,000 categories/tags
- input rate 20 items/second

Evaluation

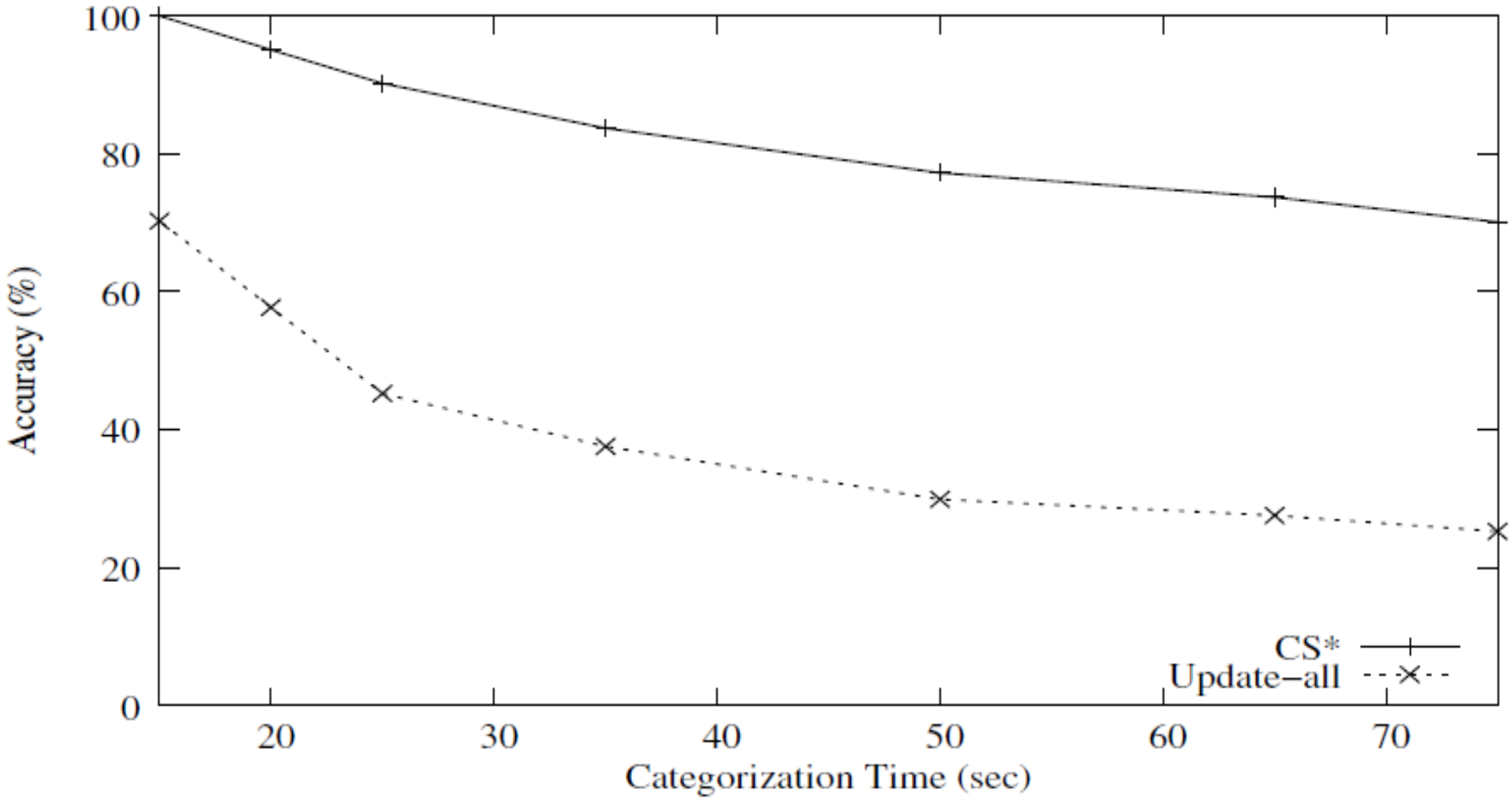
Results:

- Accuracy: $\frac{|\mathrm{Re} \cap \mathrm{Re}'|}{K}$
- Re: result-set of categories of CS*
- Re’: optimal result-set of categories
- K: # of categories

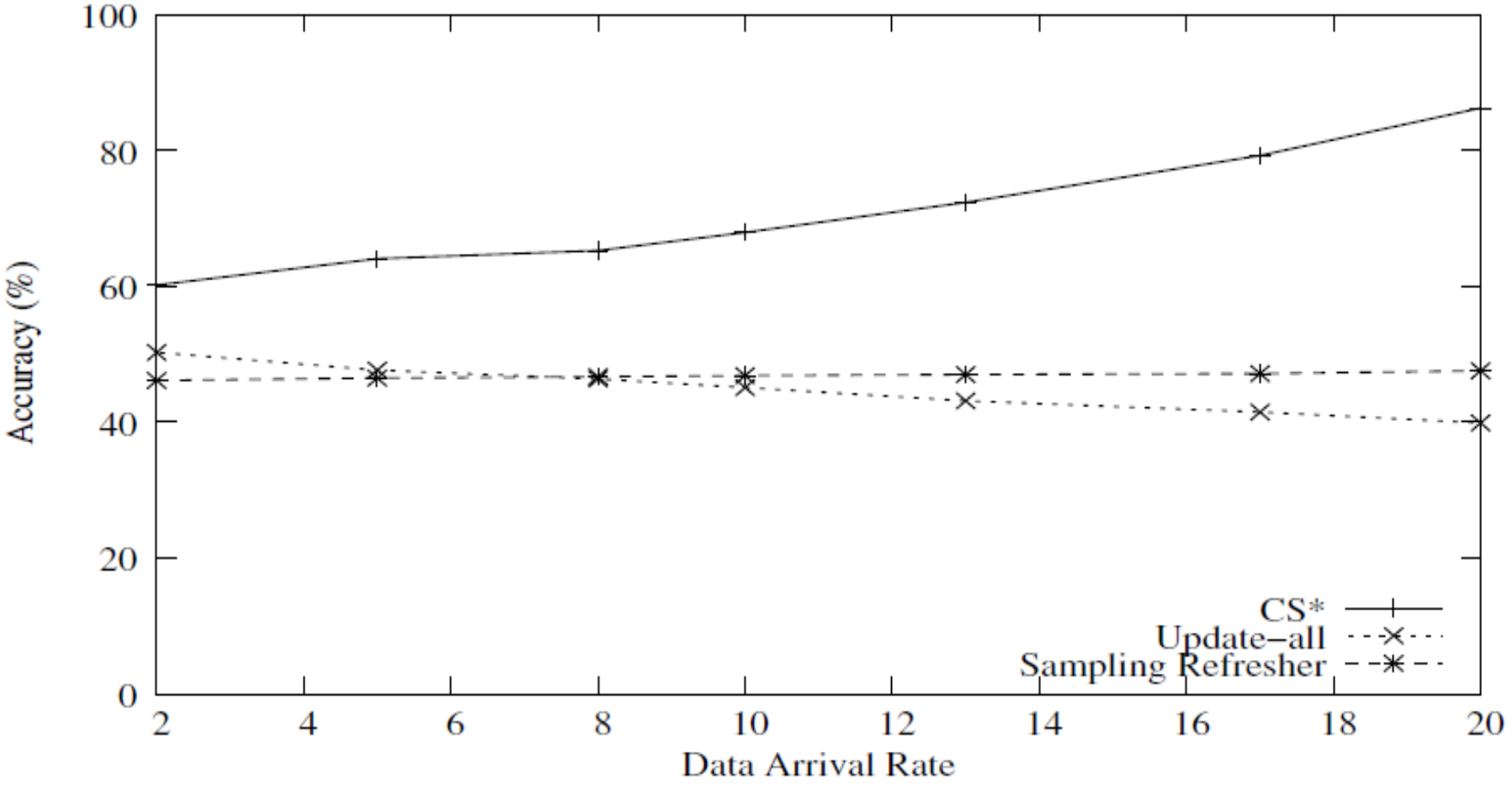
Evaluation


Accuracy (%)
Processing Power
0
20
40
60
80
100
50
100
150
200
250
300
350
400
450
500
CS* (25K)
CS* (50K)
CS* (100K)
Update–all (25K)
Update–all (50K)
Update–all (100K)

Evaluation


100
80
60
40
20
0
Accuracy (%)
20
30
40
50
60
70
Categorization Time (sec)
CS*
Update–all

Evaluation


100
80
60
40
20
0
Accuracy (%)
2
4
6
8
10
12
14
16
18
20
Data Arrival Rate
CS*
Update–all
Sampling Refresher

Evaluation

α	Categorization Cost	Processing Power		Extra Power Required
		CS*	Update-all	
20	25	300	493	64.33%
20	50	594	982	65.31%
10	25	155	244	57.42%

- Query answering module using 2-level TA only using 20% of categories for top-K

Outline

- Motivation
- Ranking categories
- Maintaining data
- Answer Queries
- Evaluation
- **Conclusion**

Conclusion

- CS* system solves problem of enabling keyword search on changing data
- Consists of two components
- Highly achieved accuracy, with high scalability, able to handle large number of data items

Thank you!