Efficiently Querying Natural Language Text

Davood Rafiei
University of Alberta

Collaborators

Haobin Li

Pirooz Chubak

Eddie Santos
Stephen Romansky
James Moore

Christopher Pinchak

Dekang Lin

Data Model

- Merril Lynch rule
  - "80 percent or more of corporate information is locked in e-mail, documents, audio, …"
- 3Vs (volume, variety, velocity)
- Move toward human readable/interpretable data models
  - Such as natural language text

Impact of less invasive treatments including sclerotherapy with a new agent and hemorrhoidopexy for prolapsing internal hemorrhoids.

Tokunaga Y, Sasaki H. (Int Surg. 2013)

Abstract

Abstract Conventional hemorrhoidectomy is applied for the treatment of prolapsing internal hemorrhoids. Recently, less-invasive treatments such as sclerotherapy using aluminum potassium sulphate/tannic acid (ALTA) and a procedure for prolapse and hemorrhoids (PPH) have been introduced. We compared the results of sclerotherapy with ALTA and an improved type of PPH03 with those of hemorrhoidectomy. Between January 2006 and March 2009, we performed hemorrhoidectomy in 464 patients, ALTA in 940 patients, and PPH in 148 patients with second- and third-degree internal hemorrhoids according to the Goligher's classification. The volume of ALTA injected into a hemorrhoid was 7.3 ± 2.2 (mean ± SD) mL. The duration of the operation was significantly shorter in ALTA (13 ± 2 minutes) than in hemorrhoidectomy (43 ± 5 minutes) or PPH (32 ± 12 minutes). Postoperative pain, requiring intravenous pain medications, occurred in 65 cases (14%) in hemorrhoidectomy, in 16 cases (1.7%) in ALTA, and in 1 case (0.7%) in PPH. The disappearance rates of prolapse were 100% in hemorrhoidectomy, 96% in ALTA, and 98.6% in PPH. ALTA can be performed on an outpatient basis without any severe pain or complication, and PPH is a useful alternative treatment with less pain. Less-invasive treatments are beneficial when performed with care to avoid complications.

After Stanford Named Entity Recognition

Abstract Conventional hemorrhoidectomy is applied for the treatment of prolapsing internal hemorrhoids. Recently, less-invasive treatments such as sclerotherapy using aluminum potassium sulphate/tannic acid (ALTA) and a procedure for prolapse and hemorrhoids (PPH) have been introduced. We compared the results of sclerotherapy with ALTA and an improved type of PPH03 with those of hemorrhoidectomy. Between January 2006 and March 2009, we performed hemorrhoidectomy in 464 patients, ALTA in 940 patients, and PPH in 148 patients with second- and third-degree internal hemorrhoids according to the Goligher's classification. The volume of ALTA injected into a hemorrhoid was 7.3 ± 2.2 (mean ± SD) mL. The duration of the operation was significantly shorter in ALTA (13 ± 2 minutes) than in hemorrhoidectomy (43 ± 5 minutes) or PPH (32 ± 12 minutes). Postoperative pain, requiring intravenous pain medications, occurred in 65 cases (14%) in hemorrhoidectomy, in 16 cases (1.7%) in ALTA, and in 1 case (0.7%) in PPH. The disappearance rates of prolapse were 100% in hemorrhoidectomy, 96% in ALTA, and 98.6% in PPH. ALTA can be performed on an outpatient basis without any severe pain or complication, and PPH is a useful alternative treatment with less pain. Less-invasive treatments are beneficial when performed with care to avoid complications.

Organization
Percent
Date

After (manual) information extraction

```
<?xml version="1.0" encoding="UTF-8"?>
<study>
 <disease> prolapsing internal hemorrhoid </disease>
 <treatments>
  <treatment><name>Conventional hemorrhoidectomy</name></treatment>
  <treatment><name>sclerotherapy using aluminum potassium sulphate/tannic acid</name><abbrv> sclerotherapy usi
  <treatment><name>procedure for prolapse and hemorrhoids</name><abbrv>PPH</abbrv></treatment>
 </treatments>
 <compared>
  <treatment><name>sclerotherapy with ALTA</name>
       <patientCnt>940</patientCnt>
       <duration>13+-2</duration>
       <painmed>16</painmed>
       <disappearanceRate>96</disappearanceRate>
  </treatment>
  <treatment><name>PPH03</name>
       <patientCnt>148</patientCnt>
       <duration>32+-12</duration>
       <painmed>1</painmed>
       <disappearanceRate>98.6</disappearanceRate>
  </treatment>
  <treatment><name>hemorrhoidectomy</name>
        <patientCnt>484</patientCnt>
       <duration>43+-5</duration>
       <painmed>65</painmed>
       <disappearanceRate>100</disappearanceRate>
  </treatment>
 </compared>
</study>
```

NATURAL LANGUAGE TEXT AS A DATA MODEL

Related work

- Text documents
  - Keyword search
  - Document/snippet granularity
- Dictionary
  - OED project at Waterloo (F. Tompa et al., 1984-1994?)
- Information extraction
  - e.g. KnowItAll (2004), TextRunner (2007), …
  - Other entity or relationship extractions

Question Answering (QA)

- Components
  - Question analysis
  - Information retrieval
  - Answer extraction
  - Answer typing/verification
- Progress
  - TREC QA track, 1999-2005
    - Language computer Corp. (#1 for 2002-2005)
  - IBM Watson, 2011

Question Answering (Cont.)

- Questions vs. queries
  - Additional effort to understand
  - Sometimes ambiguous
- Scalability
  - Usually domain-dependent
  - Speed and cost
- Orthogonal to ours

Other related work

- E. F. Codd. Seven steps to rendezvous with the casual user. In IFIP Working Conference on Data Base Management, pages 179–200, 1974

Modeling Nat. Lang. Text

- A sequence of tokens

- A (parse) tree

Outline

- A sequence of tokens
  - Wild card queries
  - Query rewriting and ranking
  - Indexing for wild card queries
  - Evaluation
- A (parse) tree
  - Indexing for tree pattern queries
  - Query decomposition
  - Evaluation
- Conclusions

Wild Card Queries

[Li & Rafiei, SIGIR 2006, CIKM 2009]

- % is the prime minister of Canada
- % invented the light bulb
- % invented %
- % is a summer *blockbuster*

DeWild

Data Extraction Using Wild Card

% is a car manufacturer

need help?

Search

Examples: car manufacturers | countries | summer blockbusters | Who invented the light bulb?

Instance	Weight
general motors	0.216994
toyota	0.196666
hyundai	0.194849
ford	0.19083
gm	0.19083
audi	0.188238
honda	0.186772
daimler chrysler	0.160607

Text Patterns

- Data wrapped in text patterns
  - <name> was born in <year>
  - Also referred to as surface text patterns [Ravichandran and Hovy, 2002]
- Queries ~ text patterns
- Issues
  - Rewriting relationships between patterns
  - Ranking

Rewriting Rule Language

- Express different ways of rewriting a query
- Rules *text-pattern → rewriting*

- Exhaust all matching rules
  - to obtain rewritings

Rewriting Rules

- Hyponym patterns [Hearst, 1992]
  - X such as Y
  - X including Y
  - Y and other X
- Morphological patterns
  - X invents Y
  - Y is invented by X
- Specific patterns
  - X discovers Y
  - X finds Y
  - X stumbles upon Y

Rewriting Rules (Cont.)

```
# nopos
(.+),? such as (.+)
such (.+) as (.+)
(.+),? especially (.+)
(.+),? including (.+)
->
$1 such as $2          && noun(,$1)
such $1 as $2          && noun(,$1)
$1, especially $2      && noun(,$1)
$1, including $2       && noun(,$1)
$2, and other $1       && noun(,$1)
$2, or other $1        && noun(,$1)
$2, a $1               && noun($1,)
$2 is a $1             && noun($1,)
```

```
#pos
N<([^<>]+)>N,? V<(\w+)>V by N<([^<>]+)>N
N<([^<>]+)>N V<is (\w+)>V by N<([^<>]+)>N
N<([^<>]+)>N V<are (\w+)>V by N<([^<>]+)>N
N<([^<>]+)>N V<was (\w+)>V by N<([^<>]+)>N
N<([^<>]+)>N V<were (\w+)>V by N<([^<>]+)>N
->
$3 $2 $1               && verb($2,,,)
$3 $2 $1               && verb(,$2,,)
$3 $2 $1               && verb(,,$2,)
$3 will $2 $1          && verb($2,,,)
$3 is going to $2 $1   && verb($2,,,)
$1 is $2 by $3         && verb(,,,$2)
$1 was $2 by $3        && verb(,,,$2)
$1 are $2 by $3        && verb(,,,$2)
```

```
noun(country, countries)
```

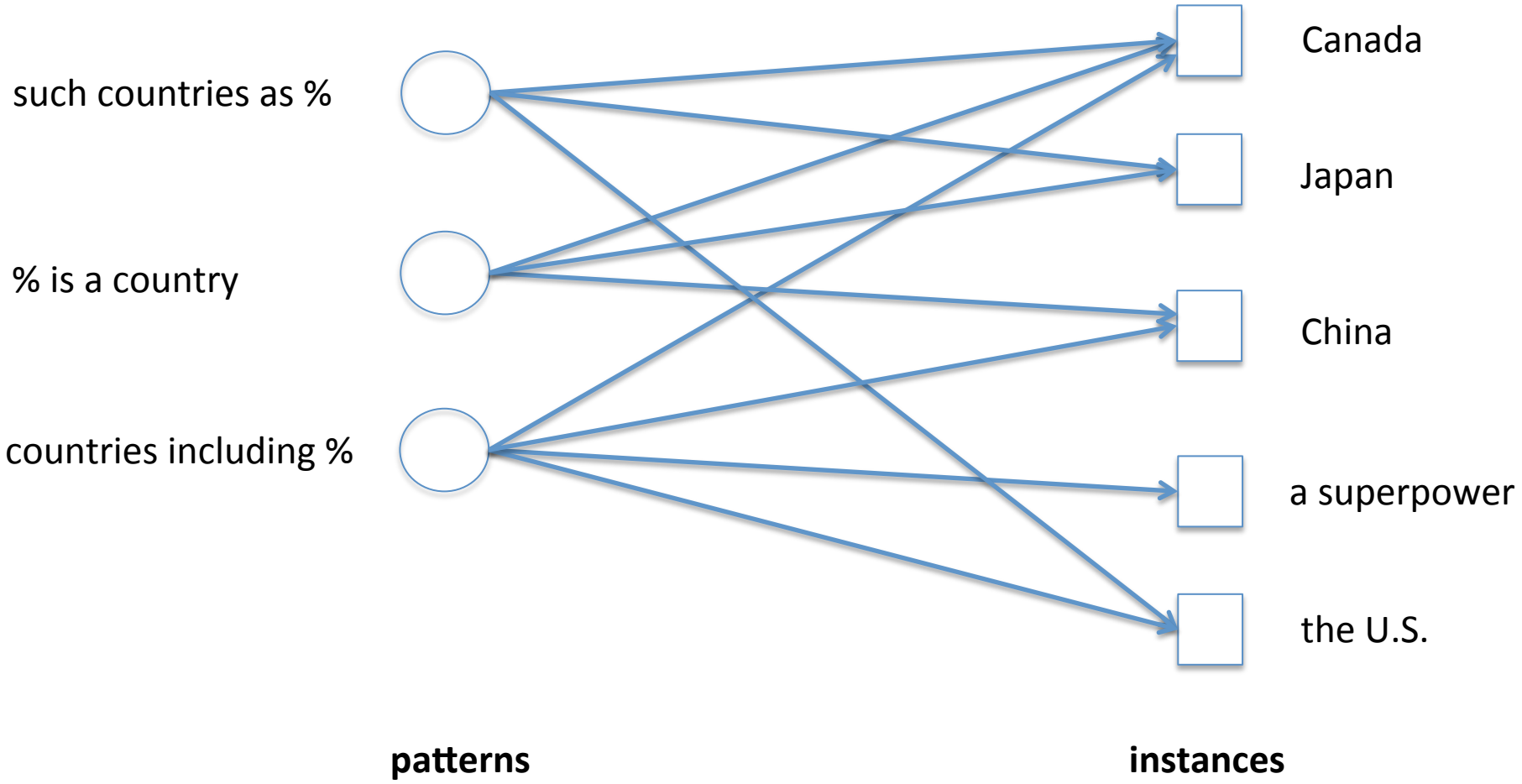
```
verb(go, goes, went, gone)
```

Ranking Heuristics

- Let t be a matching instance for q
  - NPages
    - Number of docs where t matches q or one of its rewritings
  - NPatterns
    - Number of rewritings that match t
  - Mutual Information (MI)

$$MI(q,t) = \log \frac{P(q,t)}{P(q)P(t)}$$

Mutual Reinforcing Rel.



Ranking in DeWild

- Adapted from HITS [Kleinberg, 1999]
  - a good instance is extracted by many good patterns
  - a good pattern extracts many good instances

$$WI(t) = \sum_{\{p|p \text{ extracts } t\}} WP(p)$$

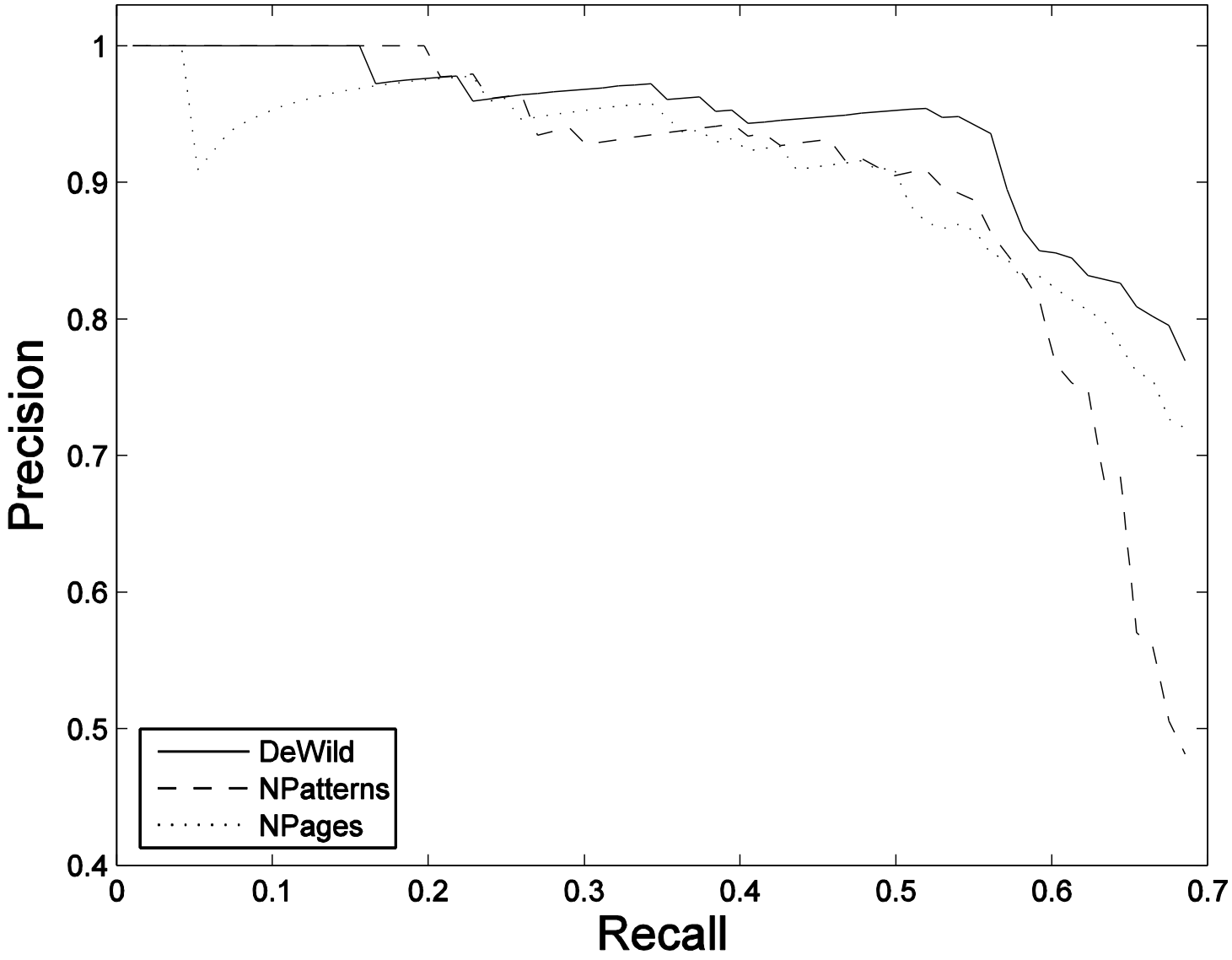
$$WP(p) = \sum_{\{t|t \text{ is extracted by } p\}} WI(t)$$

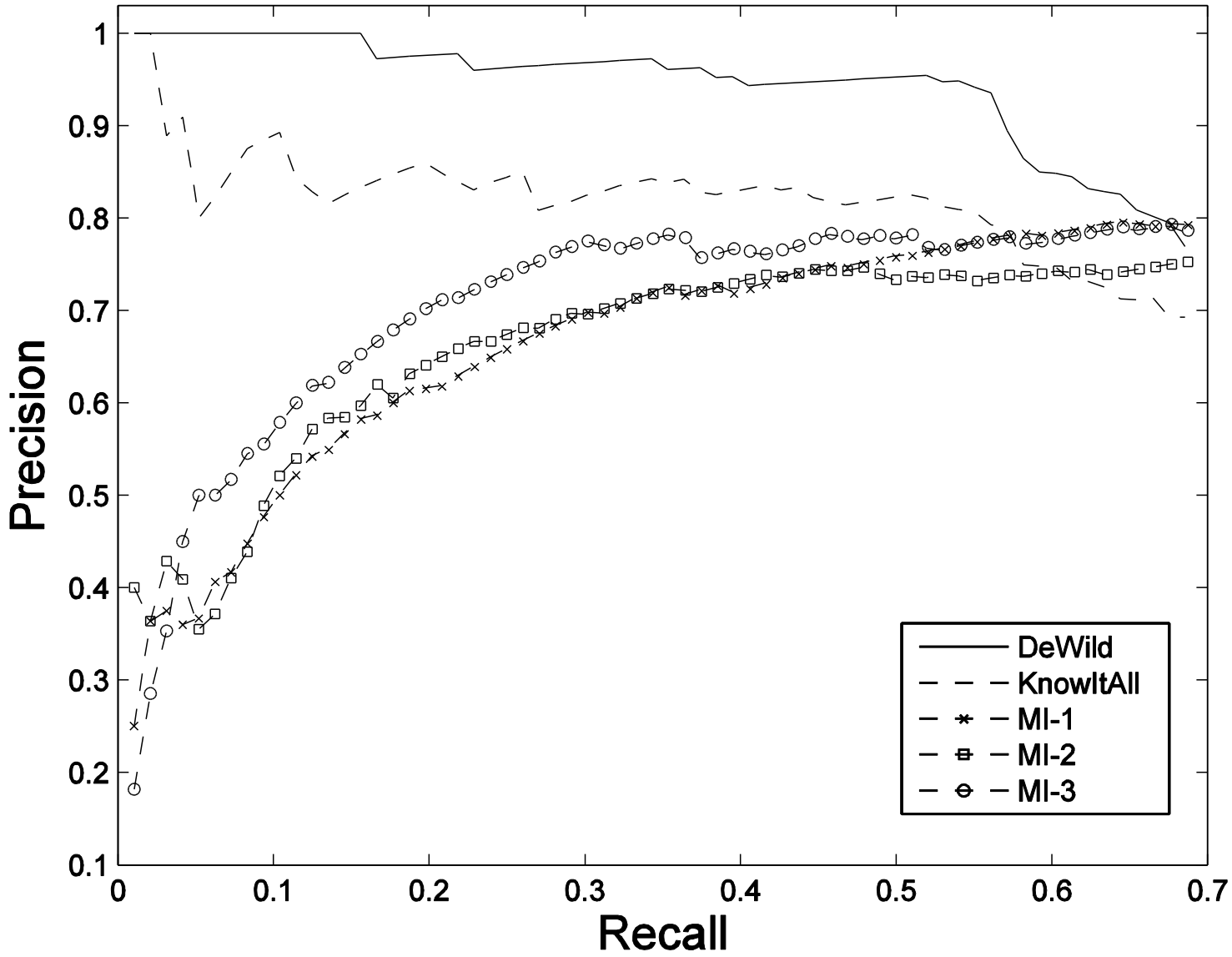
Evaluation Tasks

- Extracting a known set (country names)
  - Web as a data source
  - 200 pages for each rewriting
- QA task
- Compiling lists

Extracting Country Names

- Measure precision at each recall
- Compare to NPages, NPatterns, MI, KnowItAll
- For MI, pick the best performing rewritings
  - MI-1: “country of X”
  - MI-2: “countries such as X”
  - MI-3: “X is a country”


Precision
Recall
1
0.9
0.8
0.7
0.6
0.5
0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
DeWild
NPatterns
NPages


Precision
Recall
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
DeWild
KnowItAll
MI-1
MI-2
MI-3

QA Task (TREC 2004)

Q id	TREC ans	Our ans	overlap	# rewritings
1.1	1	2	1	3
1.2	1	na	na	na
1.3	14	5	2	1
1.4	1	na	na	na
1.5	1	4	1	1
2.1	1	2	1	1
2.2	1	4	1	1
2.3	5	7	3	1
2.4	1	7	1	11
3.1	1	3	1	1
3.2	1	na	na	na
3.3	1	na	na	na
…	…			

Compiling a list of Canadian writers

- Open-ended
- Used hand-crafted rewritings
- Over 1300 instances were extracted
- Verified the results
- Precision
  - 91 correct in the 1st 100
  - 156 correct in the 1st 200
- Of 156 correct names
  - 86 not in list A (Online Guide to Canadian Writers)
  - 70 not in list B (Canadian Literature Archive)
  - 58 not in A combined with B

Outline

- A sequence of tokens
  - Wild card queries
  - Query rewriting and ranking
  - Indexing for wild card queries
  - Evaluation
- A (parse) tree
  - Indexing for tree pattern queries
  - Query decomposition
  - Evaluation
- Conclusions

Indexing for Wild Card Queries

- Method 1: Inverted index

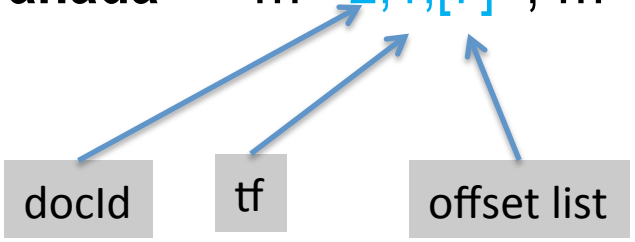
Query: Canada population is %

34,480,00 -> …, <2,1,[10]>, …

is -> <1,5,[4,16,35,58,89]>, …. <2,1,[9]>, …

population -> … <2,1,[8]> <3,1,[10]>, …

Canada -> … <2,1,[7]>, …



Indexing for Wild Card Queries (Cont.)

- Method 2: Neighbor index
  [Cafarella & Etzioni, 2005]

34,480,00 -> …, <2,1,[(10,is,-)]>, …

is -> …. <2,1,[(9,population,34,480,000)]>, …

population -> … <2,1,[(8,Canada,is)]>, …

Canada -> … <2,1,[(7,though,population)]>, …

Problems

- Long posting lists
  - Especially for terms such as ‘is’, ‘are’, ‘the’, …
- Join costs

Word Permuterm Index (WPI)

[Chubak & Rafiei, CIKM 2010]

- Three main components
  - Burrows-wheeler transformation of text (Burrows and Wheeler, 1994)
  - Structures to maintain the alphabet
  - Structures to access ranks

Word-level Burrows-wheeler transformation

- E.g. three sentences (lexicographically sorted)

  T = $ Rome is a city $ Rome is the capital of Italy $ countries such as Italy $ ~

- BW-transform
  - Find all word-level rotations of **T**
  - Sort rotations
  - The vector of the last elements is BW-transform

Word-level rotations

$ Rome is a city $ Rome is the capital of Italy $ **countries** such as Italy $ ~
Rome is a city $ Rome is the capital of Italy $ **countries** such as Italy $ ~ $
is a city $ Rome is the capital of Italy $ **countries** such as Italy $ ~ $ Rome
a city $ Rome is the capital of Italy $ **countries** such as Italy $ ~ $ Rome is
city $ Rome is the capital of Italy $ **countries** such as Italy $ ~ $ Rome is a
$ Rome is the capital of Italy $ **countries** such as Italy $ ~ $ Rome is a city
Rome is the capital of Italy $ **countries** such as Italy $ ~ $ Rome is a city $
is the capital of Italy $ **countries** such as Italy $ ~ $ Rome is a city $ Rome
the capital of Italy $ **countries** such as Italy $ ~ $ Rome is a city $ Rome is
capital of Italy $ **countries** such as Italy $ ~ $ Rome is a city $ Rome is the
of Italy $ **countries** such as Italy $ ~ $ Rome is a city $ Rome is the capital
Italy $ **countries** such as Italy $ ~ $ Rome is a city $ Rome is the capital of
$ **countries** such as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy
countries such as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $
such as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $ **countries**
as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $ **countries** such
Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $ **countries** such as
$ ~ $ Rome is a city $ Rome is the capital of Italy $ **countries** such as Italy
~ $ Rome is a city $ Rome is the capital of Italy $ **countries** such as Italy $

BW-transformation

```
1   $ Rome is a city $ Rome is the capital of Italy $ countries such as Italy $ ~
2   $ Rome is the capital of Italy $ countries such as Italy $ ~ $ Rome is a city
3   $ countries such as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy
4   $ ~ $ Rome is a city $ Rome is the capital of Italy $ countries such as Italy
5   Italy $ countries such as Italy $ ~ $ Rome is a city $ Rome is the capital of
6   Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $ countries such as
7   Rome is a city $ Rome is the capital of Italy $ countries such as Italy $ ~ $
8   Rome is the capital of Italy $ countries such as Italy $ ~ $ Rome is a city $
9   a city $ Rome is the capital of Italy $ countries such as Italy $ ~ $ Rome is
10  as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $ countries such
11  capital of Italy $ countries such as Italy $ ~ $ Rome is a city $ Rome is the
12  city $ Rome is the capital of Italy $ countries such as Italy $ ~ $ Rome is a
13  countries such as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $
14  is a city $ Rome is the capital of Italy $ countries such as Italy $ ~ $ Rome
15  is the capital of Italy $ countries such as Italy $ ~ $ Rome is a city $ Rome
16  of Italy $ countries such as Italy $ ~ $ Rome is a city $ Rome is the capital
17  such as Italy $ ~ $ Rome is a city $ Rome is the capital of Italy $ countries
18  the capital of Italy $ countries such as Italy $ ~ $ Rome is a city $ Rome is
19  ~ $ Rome is a city $ Rome is the capital of Italy $ countries such as Italy $
```

Traversing **L** backwards

i	L
1	~
2	city
3	Italy
4	Italy
5	of
6	as
7	$
8	$
9	is
10	such
11	the
12	a
13	$
14	Rome
15	Rome
16	capital
17	countries
18	is
19	$

Prev(i) = Count[L[i]] + $Rank_{L[i]}(L,i)$

Element preceding i, in L

Number elements smaller than L[i], in L

Occurrences of L[i] in the range (L[1..i])

Traversing **L** backwards

i	L
1	~
2	city
3	Italy
4	Italy
5	of
6	as
7	$
8	$
9	is
10	such
11	the
12	a
13	$
14	Rome
15	Rome
16	capital
17	countries
18	is
19	$

Prev(i) = Count[L[i]] + $\text{Rank}_{L[i]}(L,i)$

Prev(8) = Count($) + $\text{Rank}_{\$}(L,8)$
= 0 + 2 = 2
The second $ is preceded by city in **T**

Prev(10) = Count(such) + $\text{Rank}_{such}(L,10)$
= 16 + 1 = 17
such is preceded by countries in **T**

T = $ Rome is a city $ Rome is the capital of Italy $ countries such as Italy $ ~

Prev(8) Prev(10)

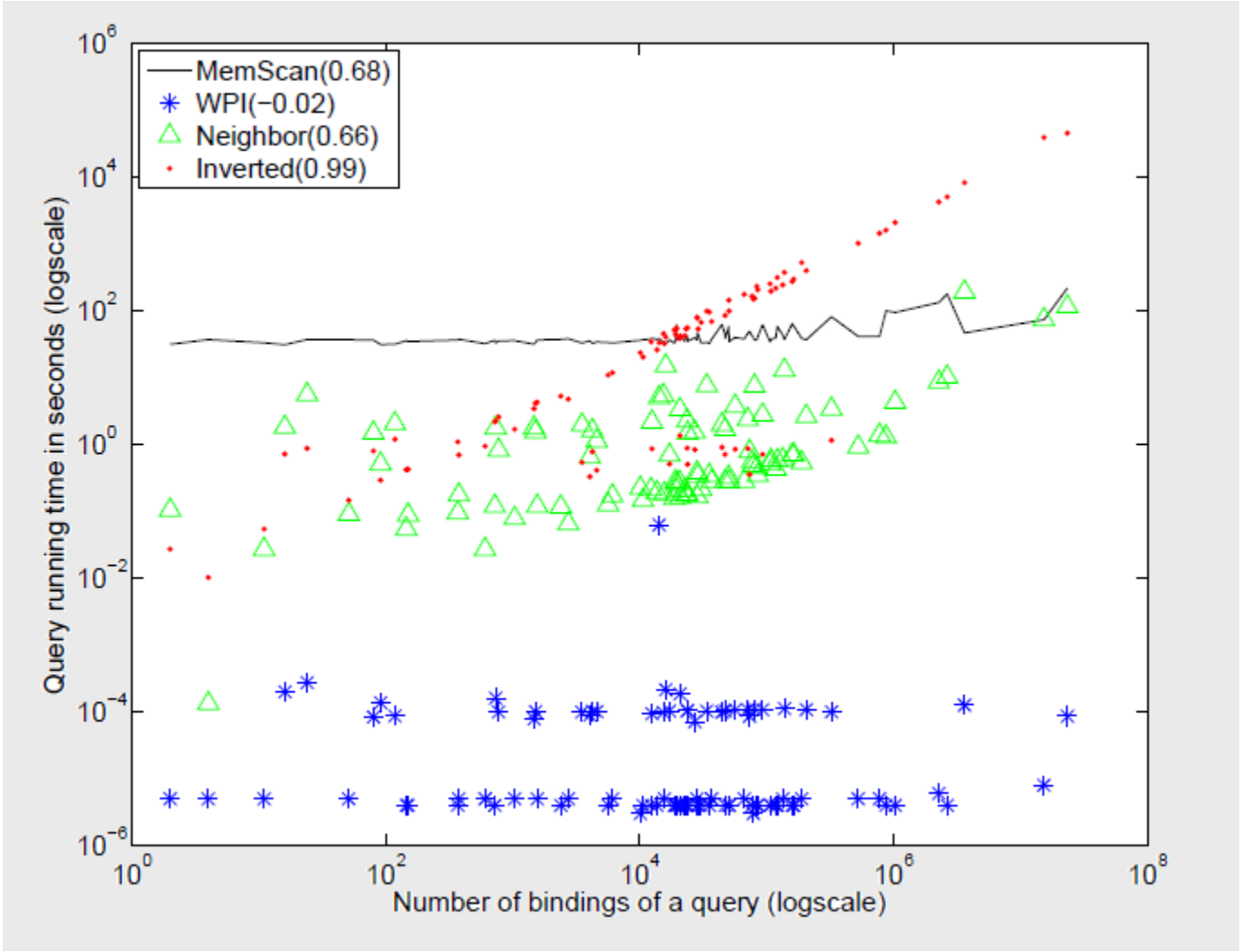
Asymptotic analysis

- Counts can be computed in constant time using a hash-table
- Ranks can be accessed in **O(log |Σ|)** time using a structure called a wavelet tree (Grossi et al., 2003)
- Each backward traversal on **L** takes **O(log |Σ|)**
- Query evaluation
  - Use query literals to backward search over **L**
  - Find the match for the wild card
  - Takes **O(m log |Σ|)** for a large number of queries
    - m = number of non-wild card words in query

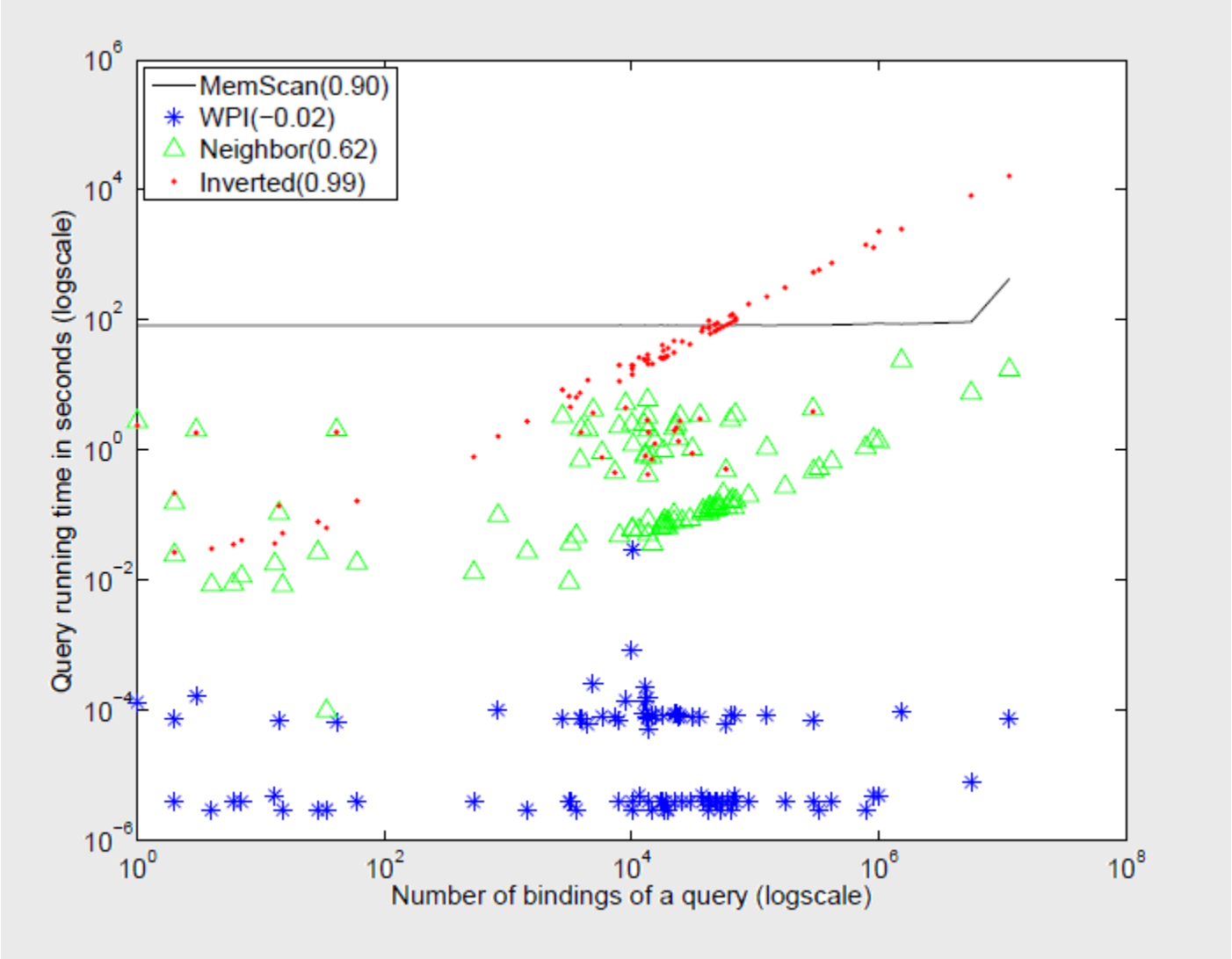
Experimental setup

- Datasets
  - 2 GBs of Sentences from a news corpus
  - 8 GBs of Documents crawled from the web
- Queries
  - Natural language questions in AOL query log
  - Subject-verb-objects obtained from a syntactic parser
  - Wild cards replaced with words in n-grams of various selectivities
- Caching
  - Assign as much cache to disk-based indexes as WPI uses memory

Running time of queries (1M documents of web data)



Running time of queries (10M sentences of news data)

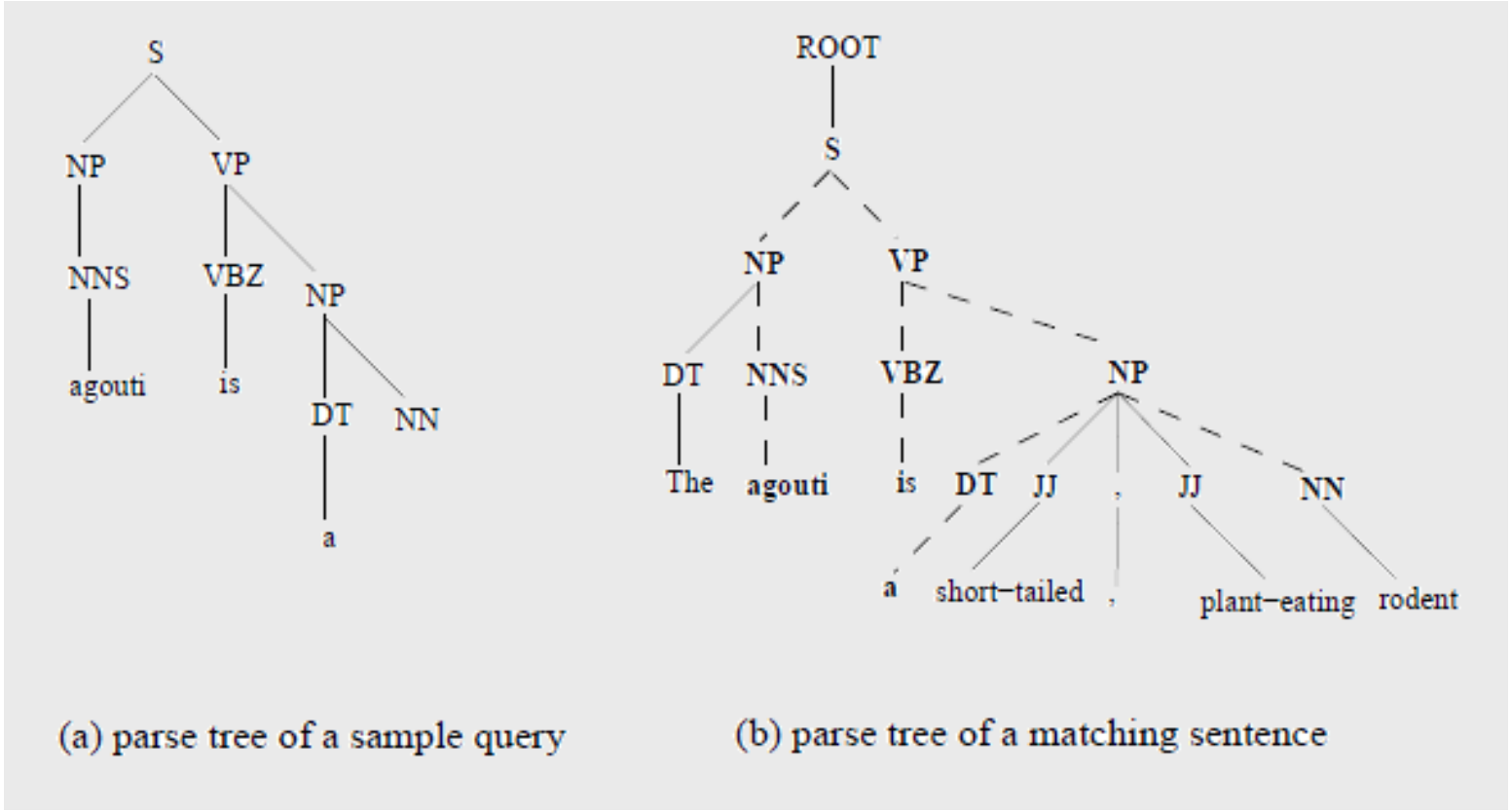


Outline

- A sequence of tokens
  - Wild card queries
  - Query rewriting and ranking
  - Indexing for wild card queries
  - Evaluation
- A (parse) tree
  - Indexing for tree pattern queries
  - Query decomposition
  - Evaluation
- Conclusions

Tree pattern queries

`What kind of animal is agouti?` (TREC-2004 QA track)



(a) parse tree of a sample query

(b) parse tree of a matching sentence

Subtree Index (SI)

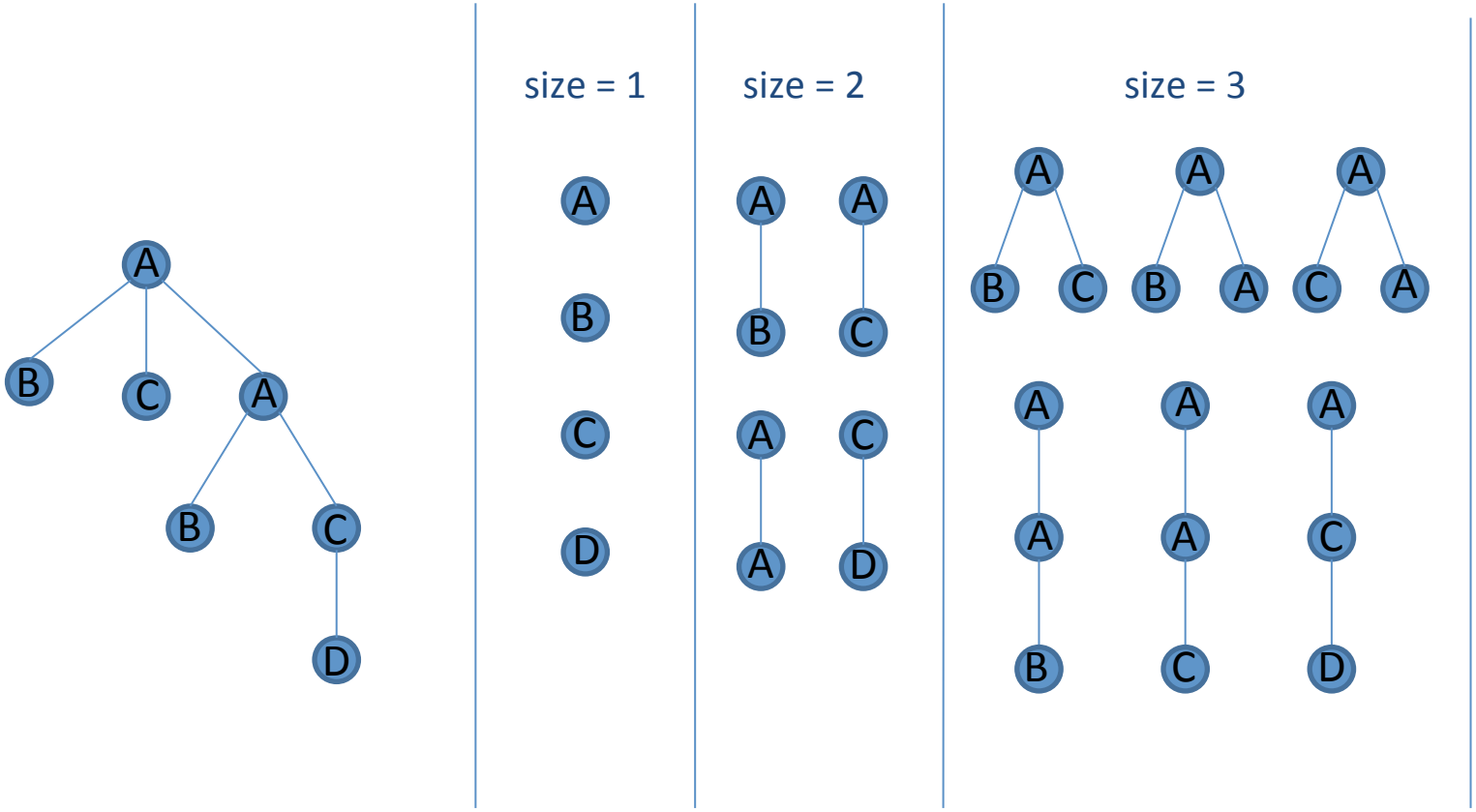
[Chubak & Rafiei, PVLDB 2012]

- Keys: unique subtrees of up to a certain size
- Posting lists: structural info. of keys

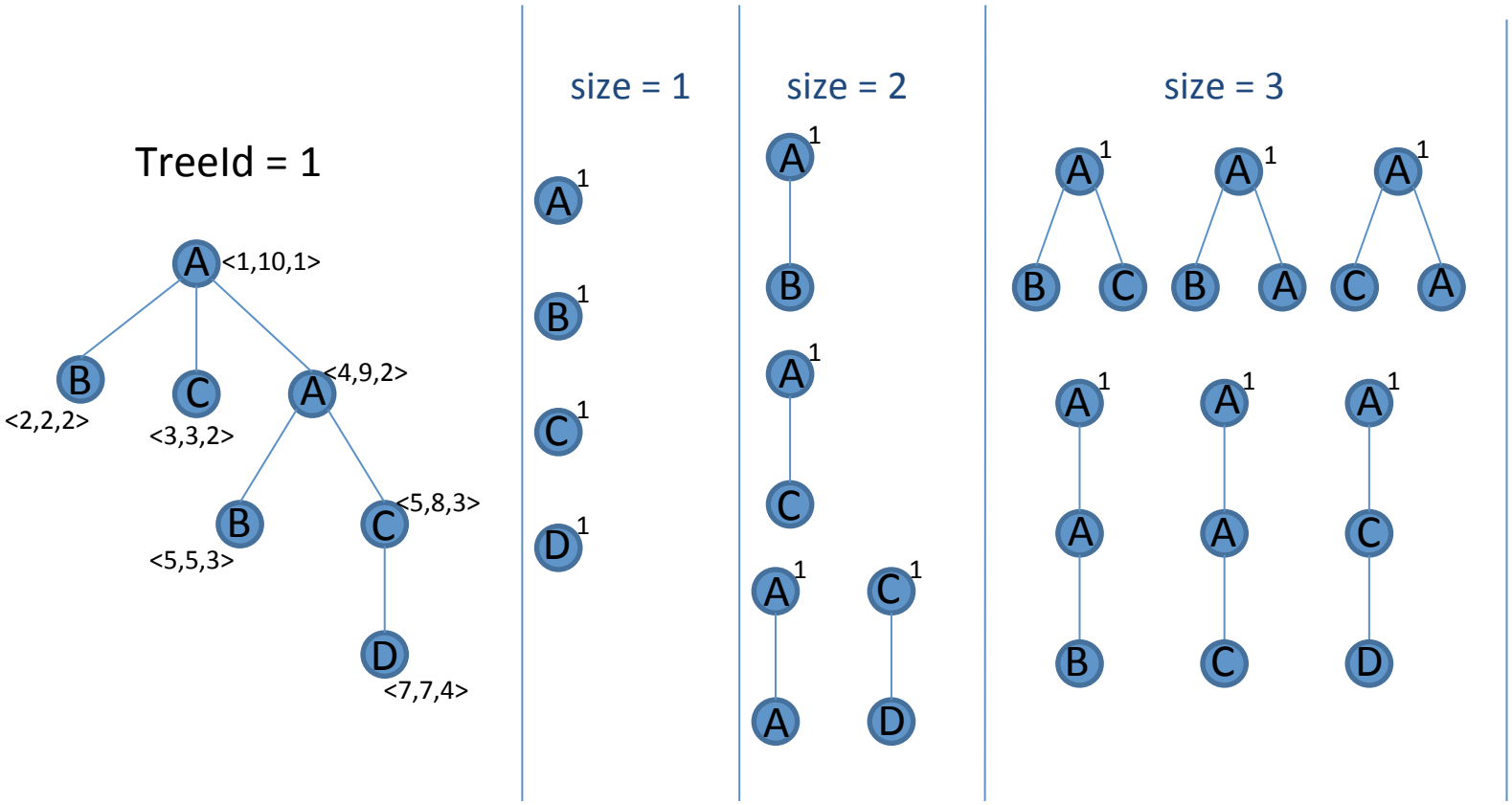
- Evaluation strategy: break queries into subtrees, fetch lists and join

- Syntactically annotated trees
  - Abundant frequent patterns -> small number of keys
  - Small average branching factor -> small number of postings

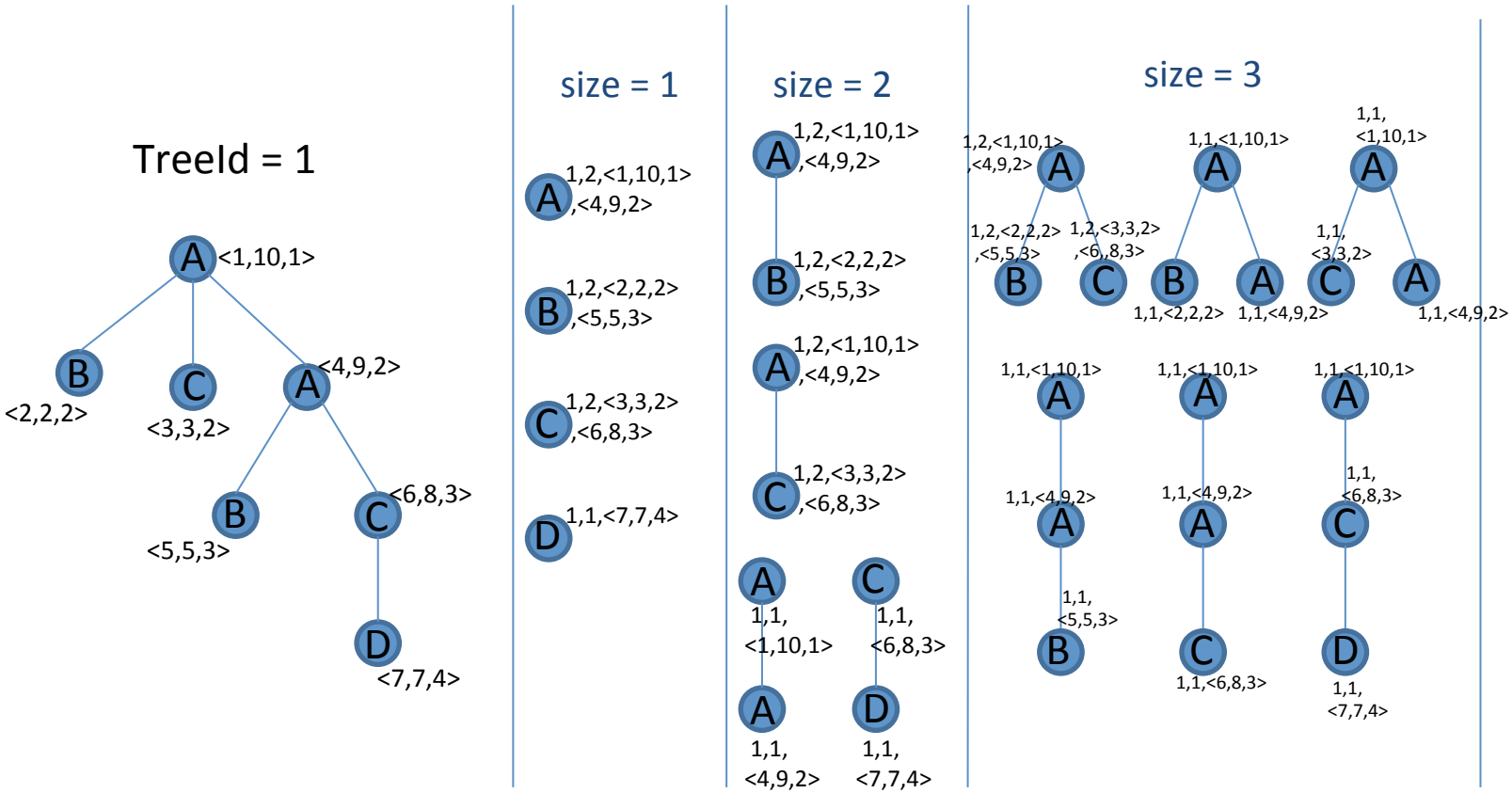
Example (subtree extraction)



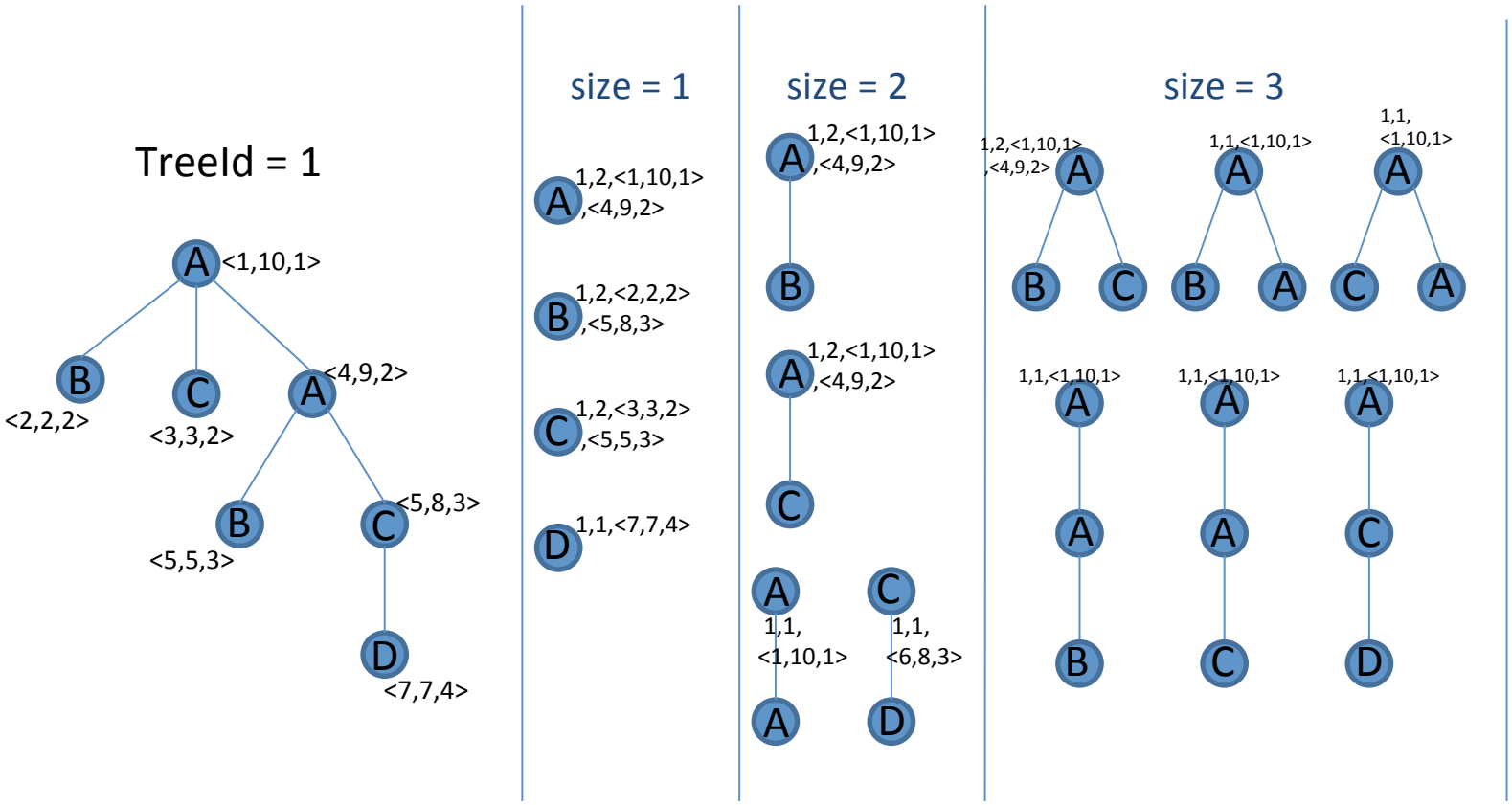
Coding Scheme (filter-based)



Coding Scheme (subtree interval)



Coding Scheme (root-split)

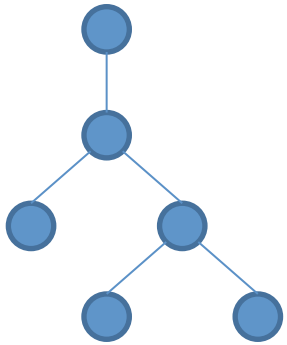


Covers

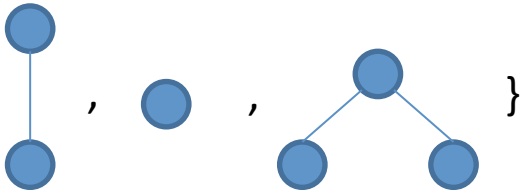
Node Cover
covers all query nodes

Example (mss=3):

Query



Node Cover = {



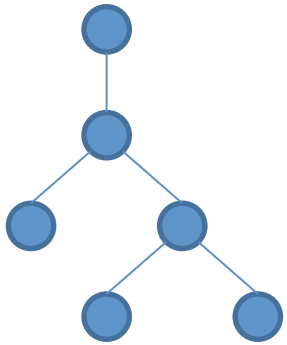
}

Max Cover

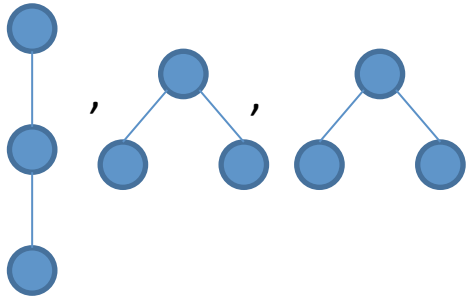
all subtree sizes = mss

Example (mss=3):

Query



Max Cover = {



,

,

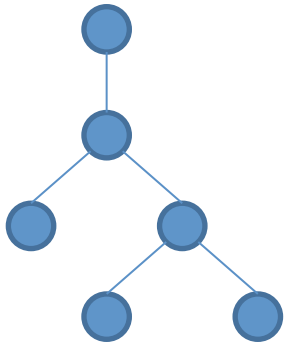
}

Root-split Cover

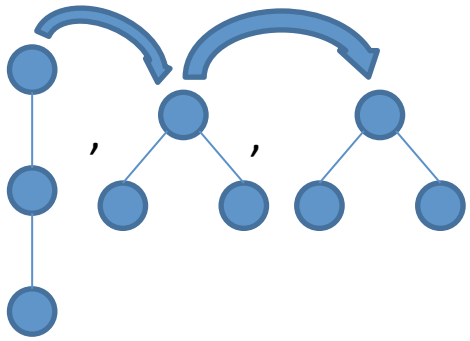
Each subtree can be root-joined with another one

Example (mss=3):

Query



Root-split Cover = {



,

,

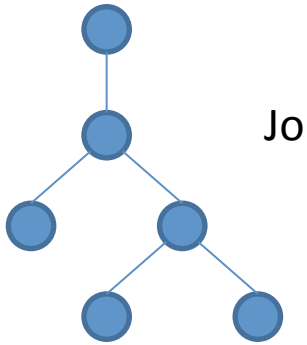
}

Join Optimality

Minimum size (max) covers

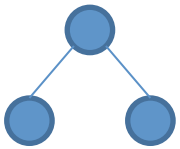
Example (mss=3):

Query



Join-optimal Cover = {

,



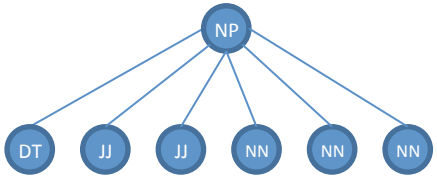
}

Query decomposition

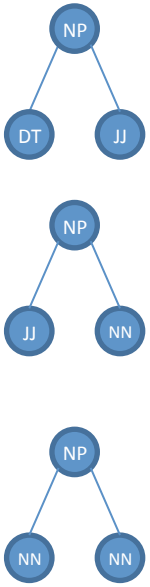
- Goal
  - Find Join-optimal covers
- Filter-based and subtree interval codings
  - an algorithm that finds join-optimal covers, if *mss <= 6*
- Root-split coding
  - an algorithm that finds the smallest root-split cover among all root-split covers

Injective Matching

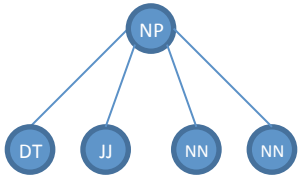
Query



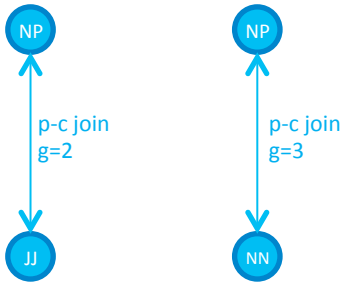
Join-optimal Cover (mss=3)



Matches



Solution:
Add the following to the list of joins
g : join granularity



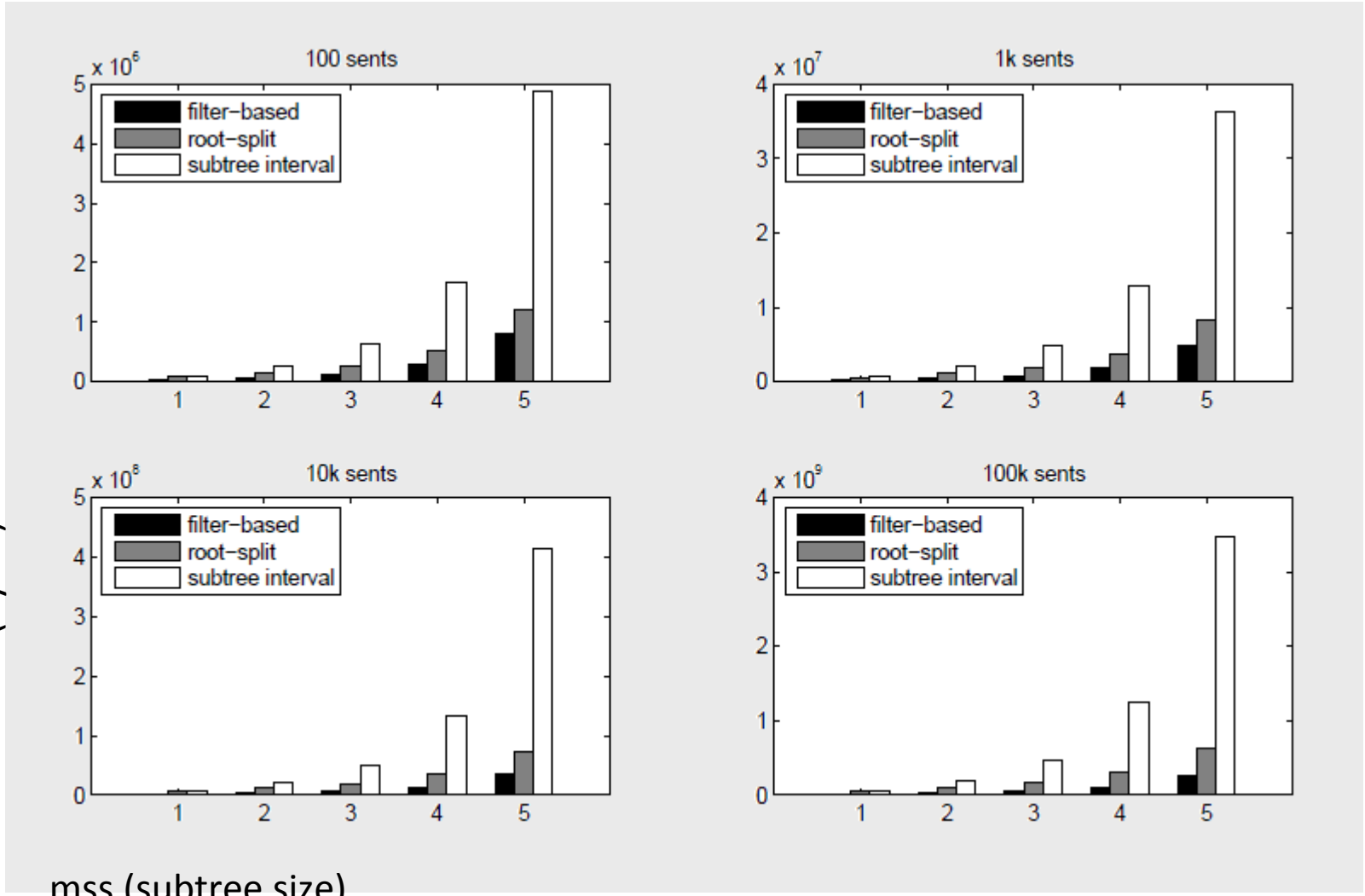
Outline

- A sequence of tokens
  - Wild card queries
  - Query rewriting and ranking
  - Indexing for wild card queries
  - Evaluation
- A (parse) tree
  - Indexing for tree pattern queries
  - Query decomposition
  - Evaluation
- Conclusions

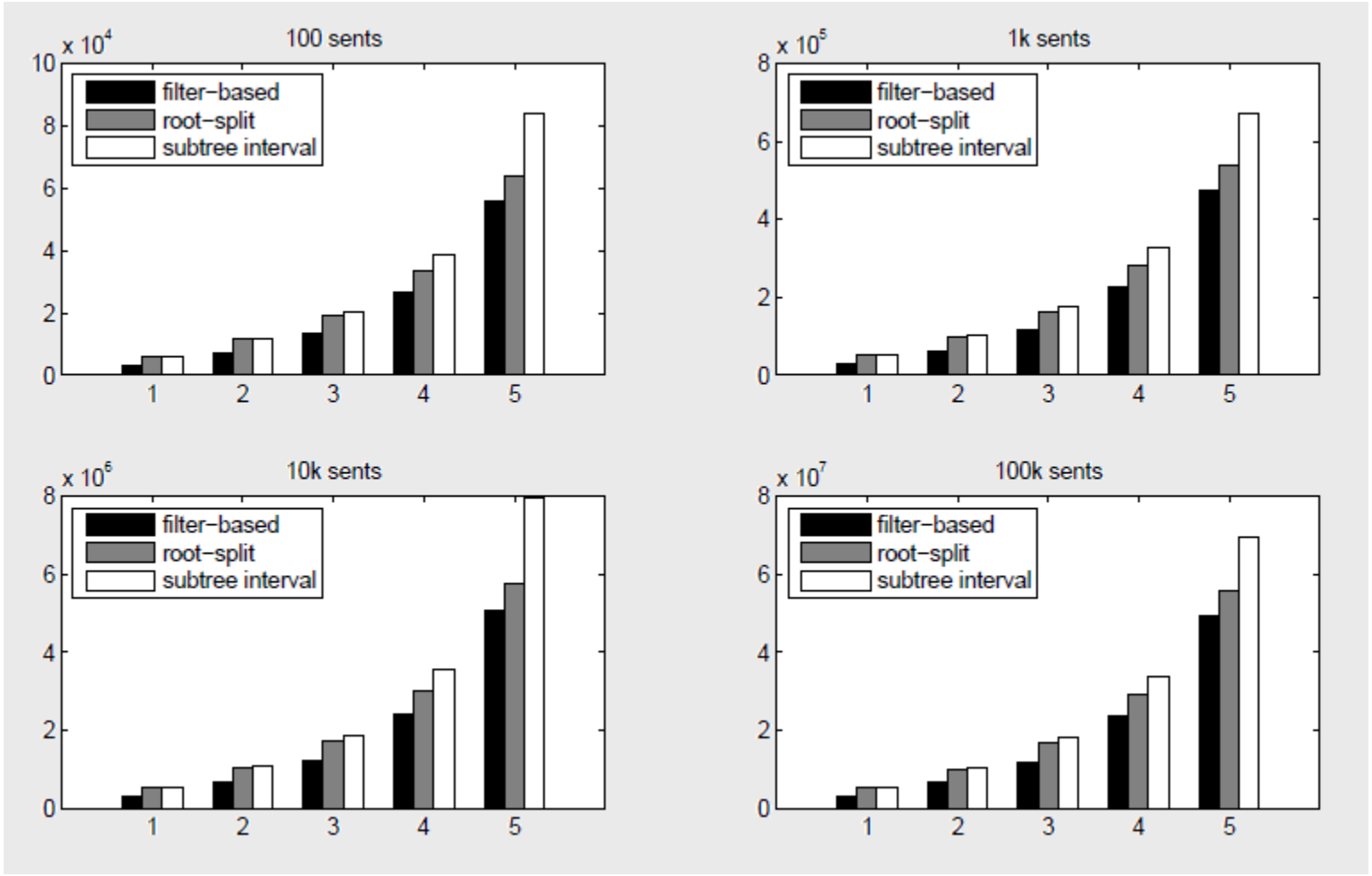
Setup

- Data
  - Parsed sentences from AQUAINT
  - Used Stanford parser
- Queries
  - WH query-set: 48 AOL-log questions (12 of what, where, which and who)
  - FB query-set: 70 subtrees with labels in different frequency classes (10 of L:low, M:medium, H:high, ML, HL, HM, HML)

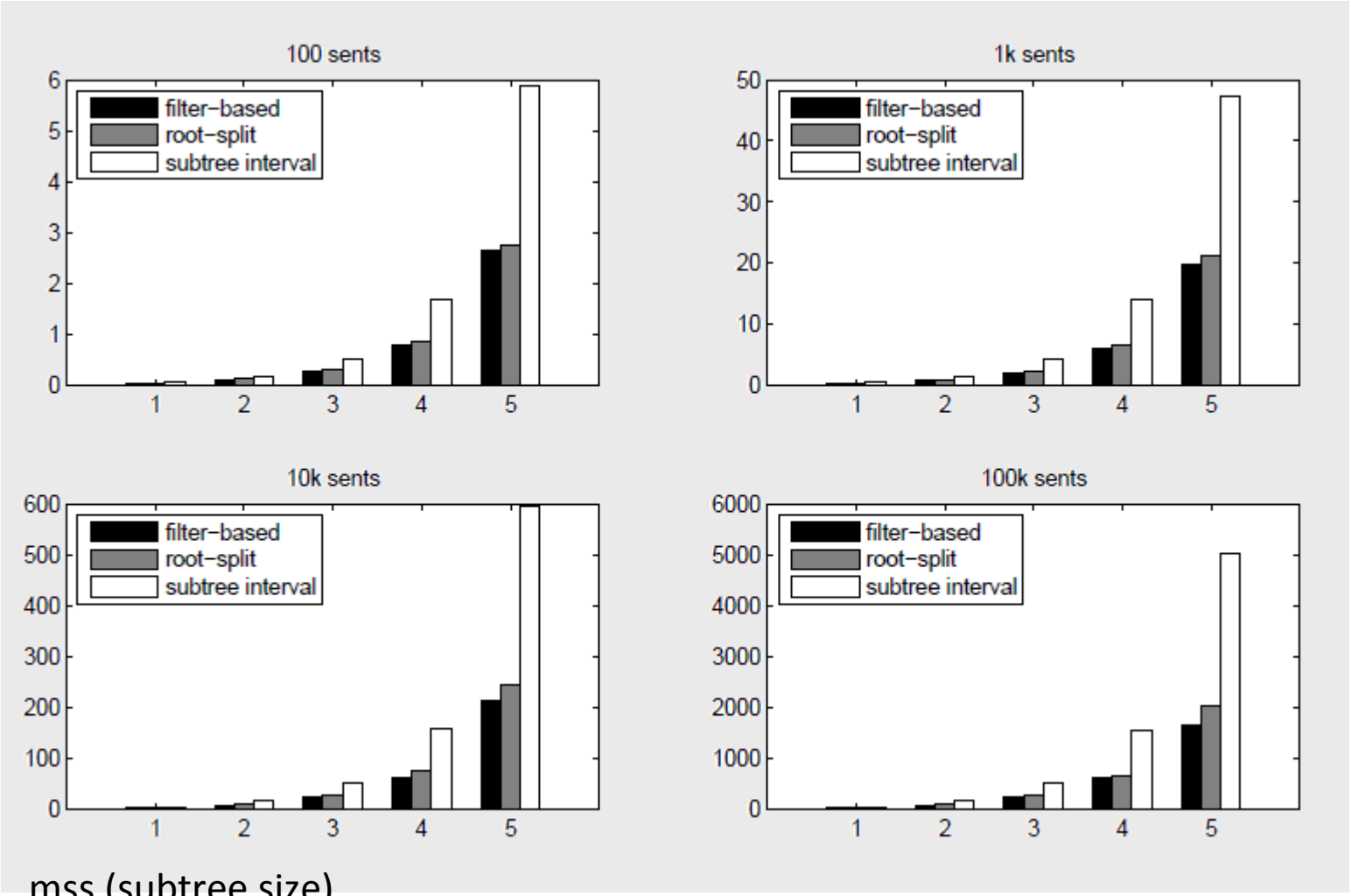
Index Size


x 10^6
100 sents
filter−based
root−split
subtree interval
x 10^7
1k sents
x 10^8
10k sents
x 10^9
100k sents
index size (bytes)
mss (subtree size)

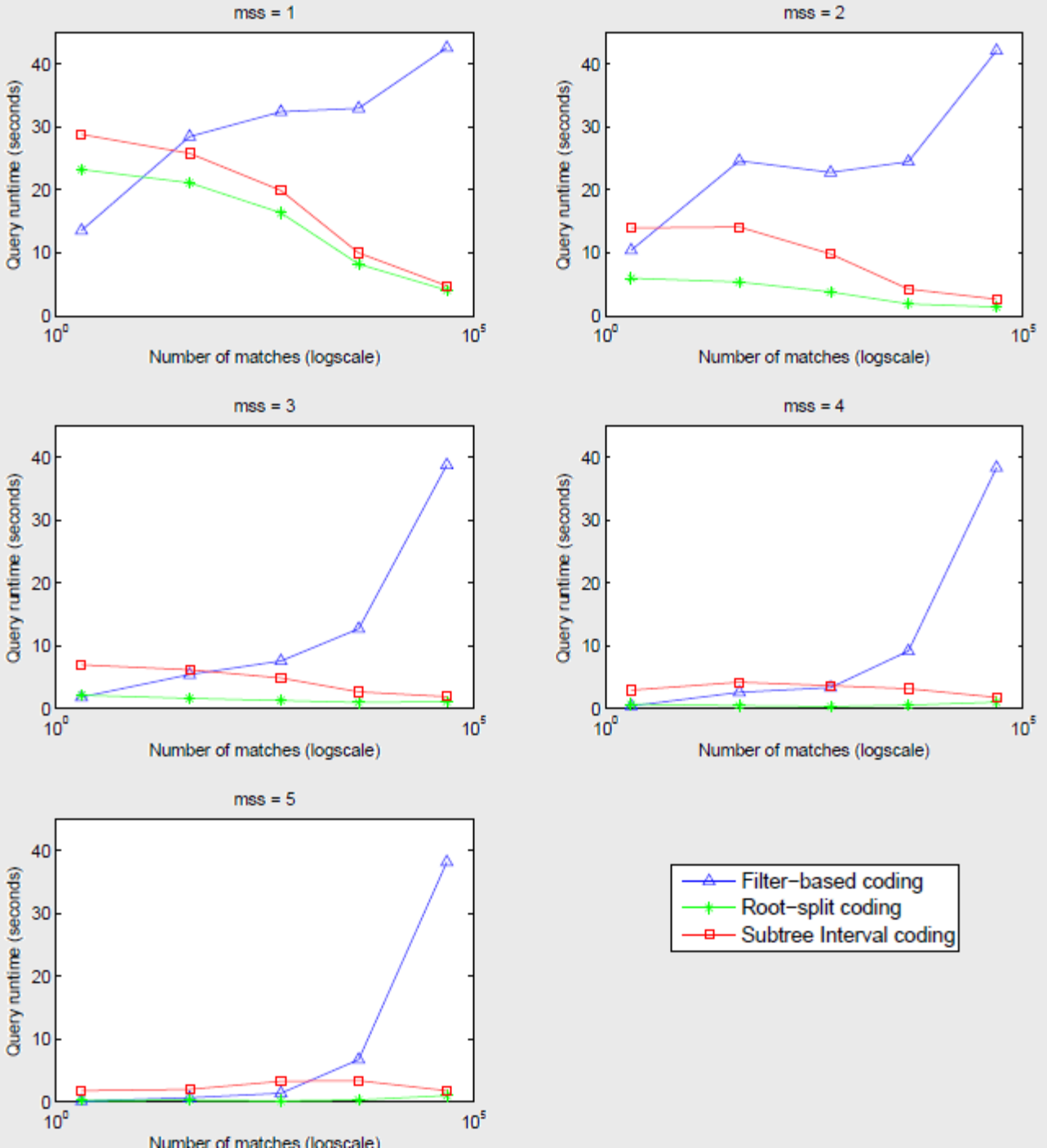
Number of postings



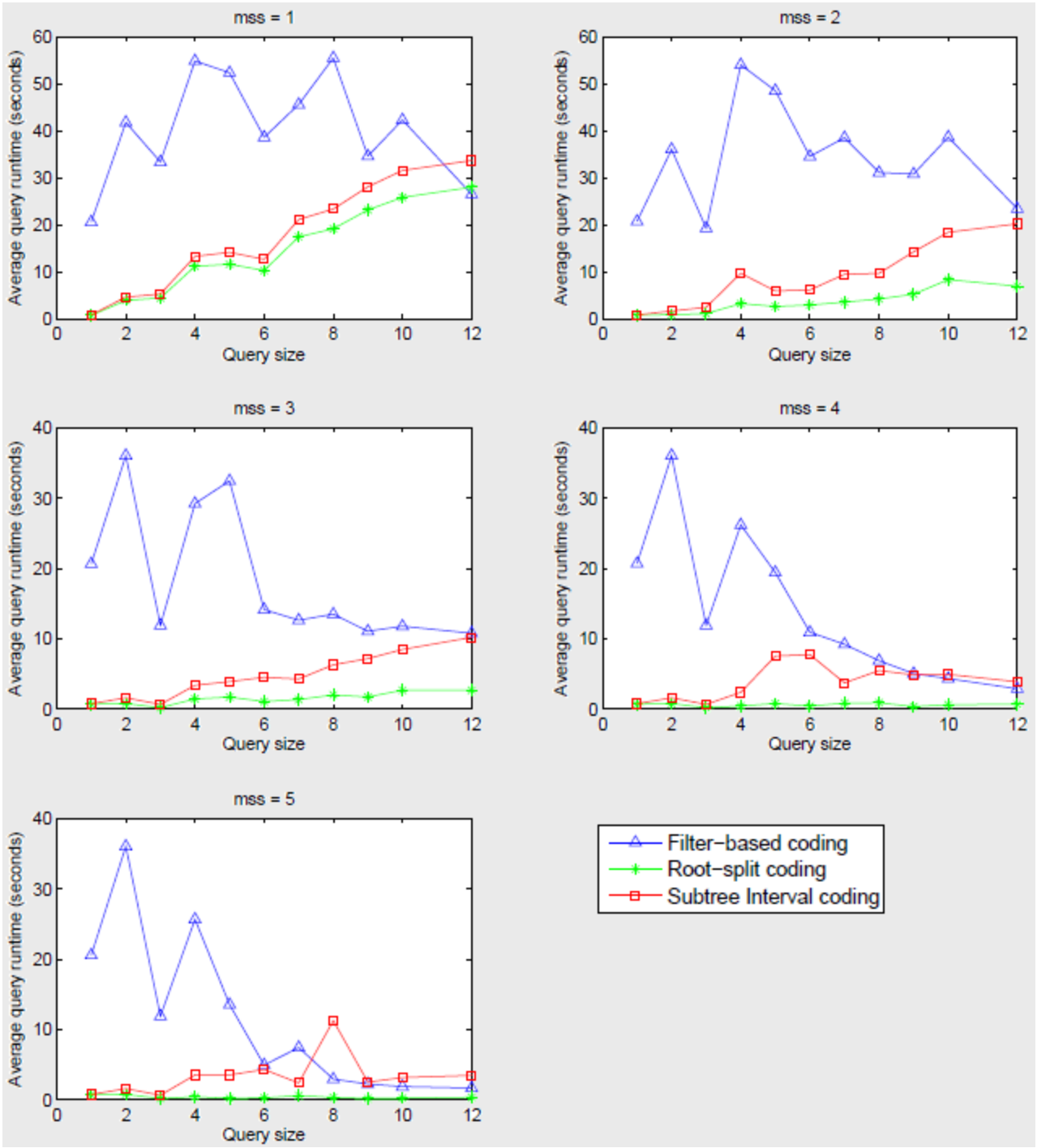
Index construction time



Querying Performance



Querying Performance



Compared with other systems (100k sentences)

Query Class	SI+RS	ATG	F-B (top 0.1%)	F-B (top 1%)	F-B (top 10%)
L	**0.09**	1.9	3.05	3.03	3.04
M	**0.01**	10.06	12.32	0.8	0.35
ML	**0.25**	2.13	10.3	9.62	9.25
H	**1.73**	22.4	39.21	34.51	34.53
HL	**1.57**	32.97	34.58	34.61	34.6
HM	**1.76**	37.08	35.54	31.40	31.57
HML	**1.76**	86.02	49.03	42.97	43.13

SI+RS = Subtree Index + root-split coding (mss=3)
ATG = ATreeGrep (Shasha et al., 2002)
FB = Frequency-based, adaptation of TreePi (Zhang et al., 2007)

Conclusions

- Natural language text as a data model
  - Allows schemaless queries
  - Scalable solutions are reachable
  - Some promising results
- Future directions
  - Rewriting rule discovery
  - More possible follow-ups

Rewriting Rule Discovery

- Automatic
  - Some progress (e.g. Ravichandran and Hovy, 2002, Lin and Pantel, 2001, Pantel et al., 2007)
  - Textual entailment challenge
- Manual
  - User has the final touch
- Crowd-Sourced

An Ongoing Work

- Phrasology
  - A two-player online game
  - Task: order rewritings from the most relevant to least

http://gamer.cs.ualberta.ca/

Other Possible Follow-ups

- WPI
  - A disk-based Word (Permuterm) Index
  - Multi-query search
- Subtree index
  - Other query decomposition and structural join algorithms