Efficient Diversity-Aware Search

Albert Angel, Nick Koudas, University of Toronto , SIGMOD 2011

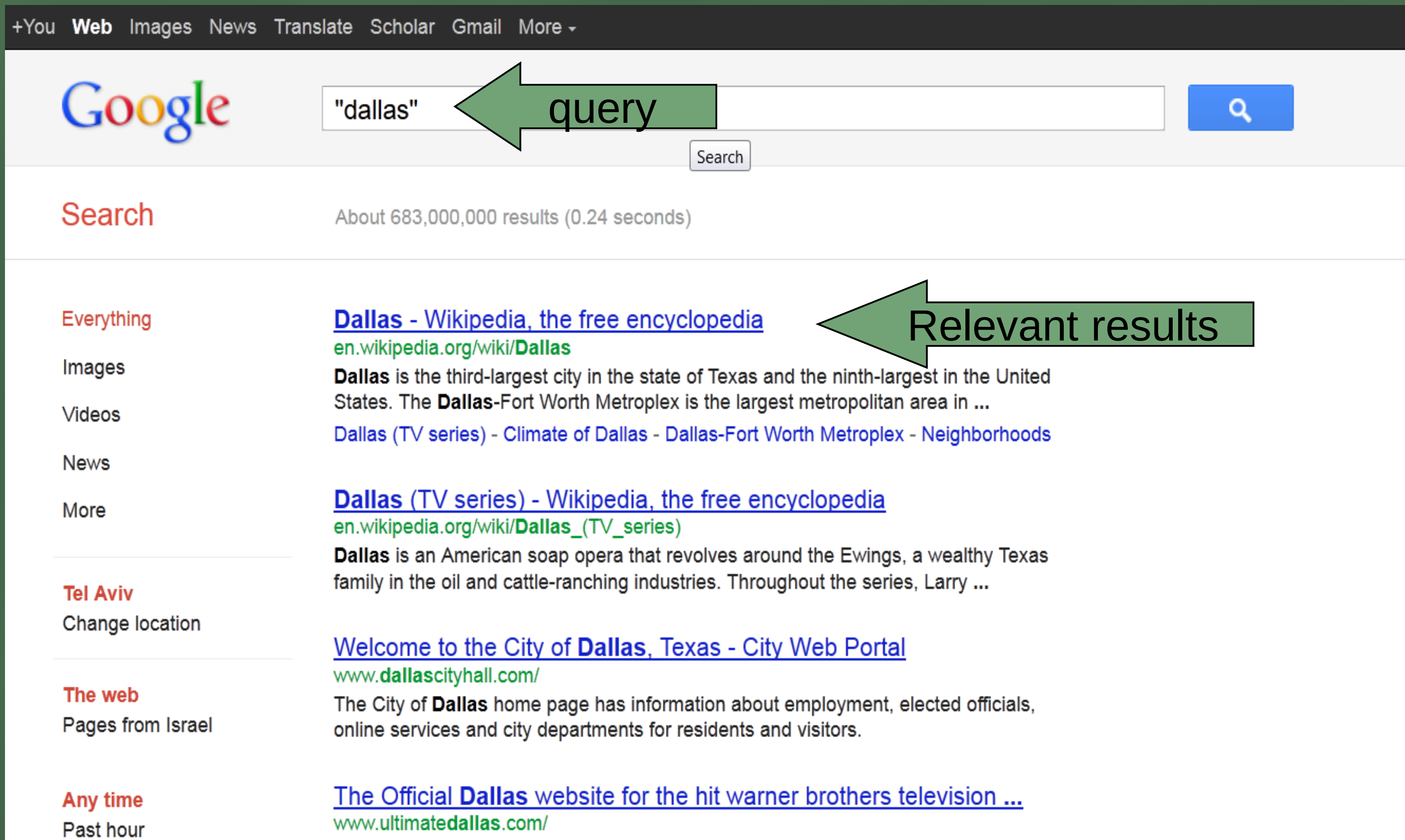
Presented by Vera Vsevolozhsky

By PresenterMedia.com

Agenda

1. Introduction, motivation
2. Problem definition and Diversity Aware Search (DAS)
3. Existing / Novel Solution (The DivGen Approach)
4. Experimental results

Info retrieval scenario



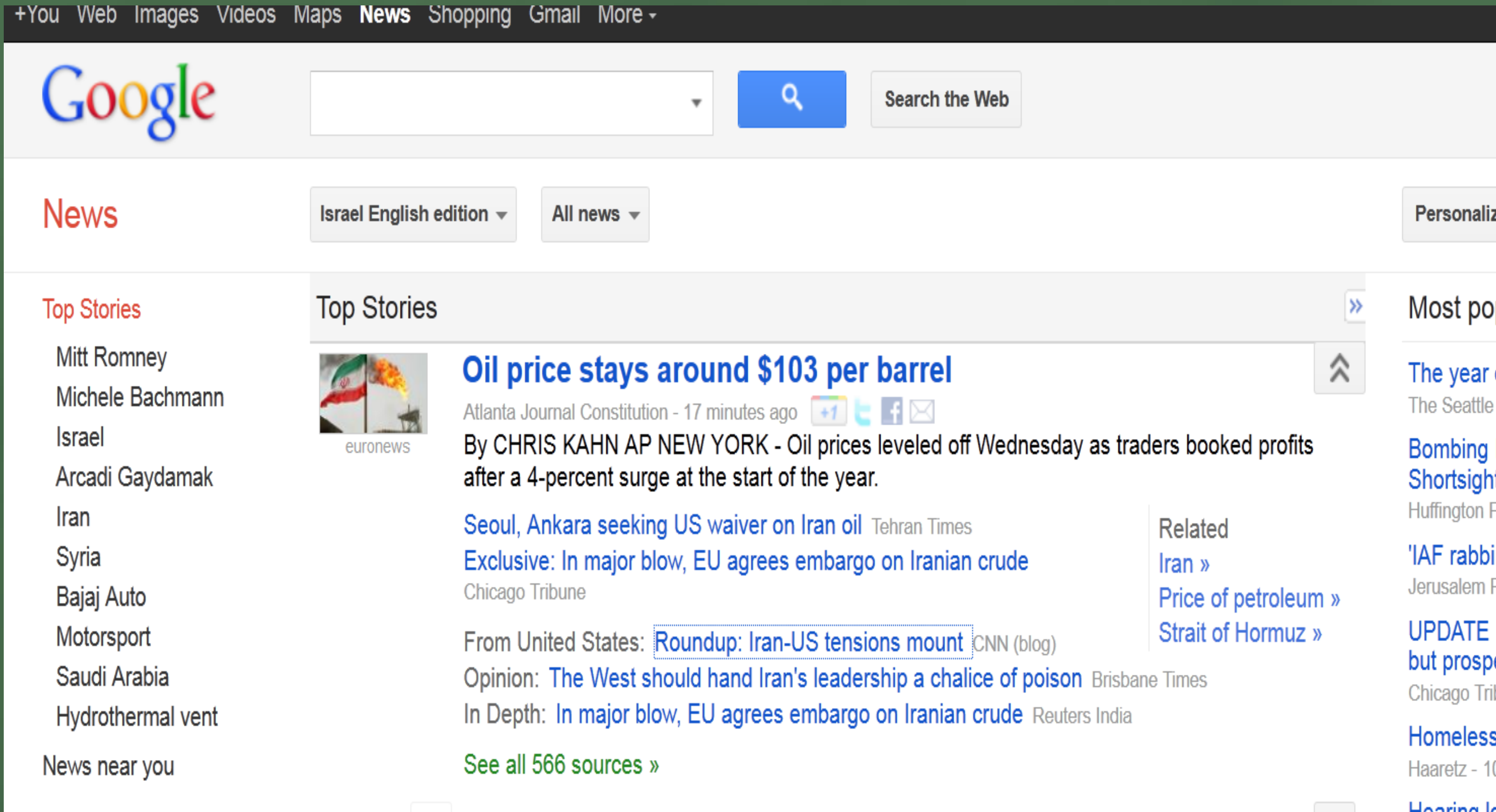
Motivation

- Does multiple query meaning may leave user unsatisfied?

- What about redundant content that is retrieved?

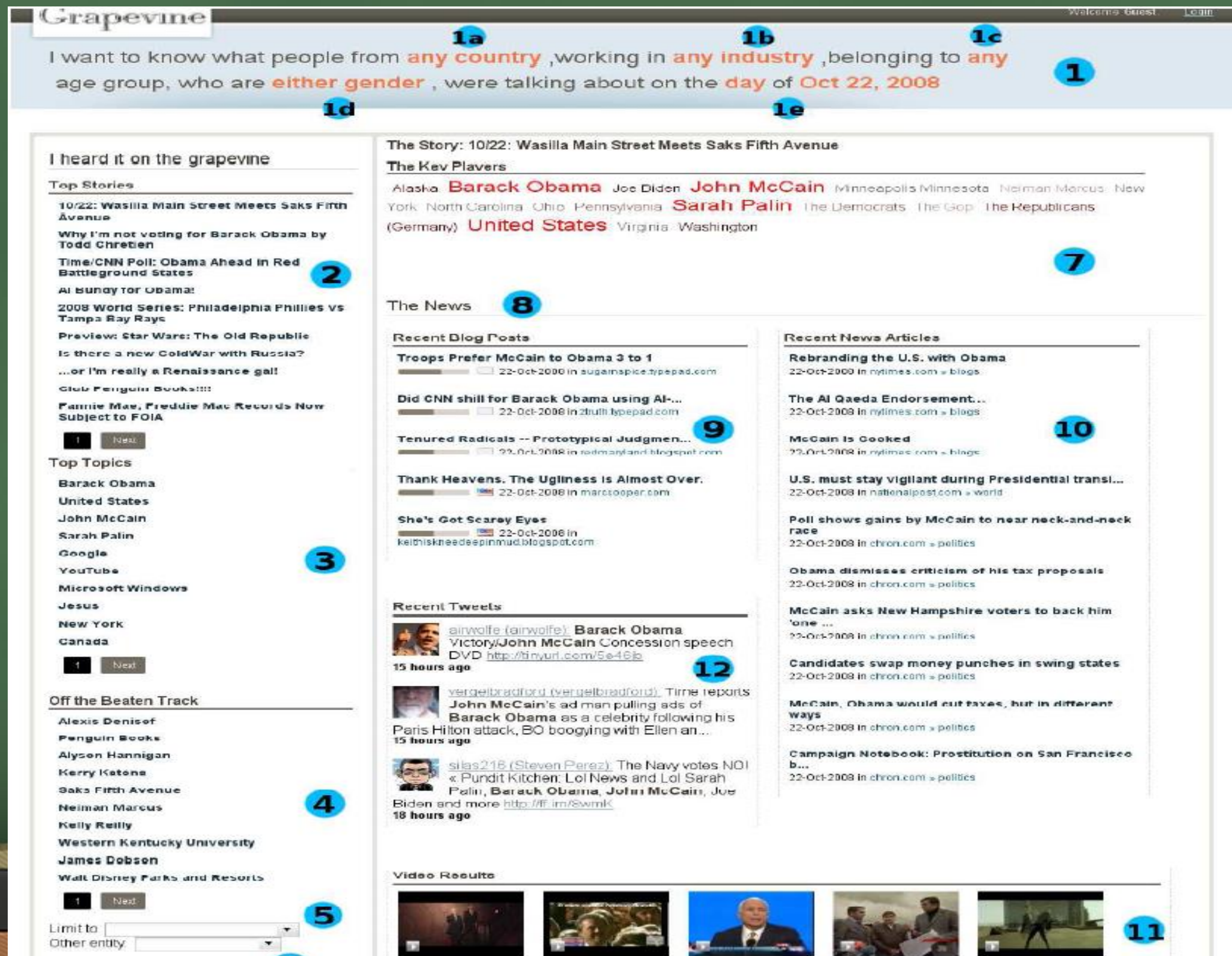
- Or maybe most users are of the exploratory nature and interesting to retrieve info that covers many aspects (diversification)?

Example 1 – Google News (news.google.com)



Example 2 – what's on Grapevine(onthegrapevine.ca)

"What are Torontonian teens talking about on blogs?"
"How is Barack Obama related to this story?"



1. Dashboard
2. Hot stories
3. Hot topics
4. On the beaten track
5. Drop-down menu
6. Auto-complete box
7. Key player
8. Content
9. Relevant blogs
10. News articles
11. Videos
12. Tweets

Diversity Aware Search (DAS)

- Data model
- User behavior
- Answer quality
- DAS definition and NP hardness

DAS – Data model

- For each document *d*:
  - **Extract** the features that describe document (text extraction):
    - Keywords in case of textual documents
    - Set of users who recommend the document in case of recommendation sys
    - Important entities
  - **Represent** the doc d as $\mathbf{d} = (d^1, d^2, \cdots)$

    where feature *i* has weight $d^i \geq 0$ in document *d*

- Represent query *q* in the same manner as document
- An answer to the query – ranked list of k documents
- Note that DAS is to return the answer whose docs are of the most use to the user

DAS - user behavior model(1)

- Document ***relevance*** can be estimated as a similarity between document *d* and query *q*:

$$\mathrm{rel}(\mathrm{d} \mid \mathrm{q}) = \mathrm{sim}(\mathrm{d}, \mathrm{q}),$$

where

$$\forall i, s.t. d_1^i > 0, d_2^i \geq d_3^i, then,$$

$$sim(d_1, d_2) \geq sim(d_1, d_3)$$

DAS - user behavior model(2)

- **Document *redundancy*** :

$$red(d \mid \{d1,\ldots,dm\}, q)$$

Then document *novelty = (1 – redundancy)* :

$$1 - red(d \mid \{d_1,\ldots,d_m\}, q) = \prod_{i=1}^{m} (1 - red(d \mid d_i, q)) = \prod_{i=1}^{m} (1 - sim(d, d_i) * f_q)$$

where d, d1,…,dm – documents, q-query and fq – focus parameter

(for example, for fq=0.4 - a document with content similar to what the user has already seen has a 40% chance of being redundant);

note that we assume that the redundancy of d wrt. d1 is independent of its redundancy wrt. to other documents

DAS - user behavior model(3)

- Document *usefulness = relevancy * novelty*

- The user goal - to locate one or more "useful" documents return as a query result

DAS - When the answer A is better than A1?

Example:

Number of document	usefulness
1	10
2	9
…	…
10	**7**
11	8

A

Is better than:

Number of document	usefulness
1	10
2	9
…	…
10	**6**
11	8

A1

Formally, **usefulness monotonicity**:

$$\exists j, s.t. \forall i \neq j, u_i^1 = u_i \text{ and } u_j^1 > u_j.$$

DAS - When the answer A is better than A1?

Example:

Number of document	usefulness
1	10
2	9
…	…
10	7
11	5

A

Is better than:

Number of document	usefulness
1	10
2	9
…	…
10	5
11	7

A1

Formally, **order of documents**:

$\exists j,l : j < l,$ such that $\forall i : i \neq j,l$ it is the case that

$u_j^1 = u_j$ and $u_j^1 = u_l > u_l^1 = u_j$.

DAS - definition

- DEFINITION 3.2. Given a definition of answer quality, and provides a (non-strict) total ordering of answers,

  DAS is the problem of finding an answer to a query such that there does not exist another answer of higher quality.

DAS –NP hardness

- **THEOREM 3.3.** In the general case, as per def. 3.2, DAS is **NP-hard**

  Proof - reduction from independent set

Restriction on DAS - Strict Order Dominance

Example:

Number of document	usefulness
1	10
2	9
…	…
10	7
11	6

A'

Is preferable than:

Number of document	usefulness
1	10
2	9
…	…
10	6
11	5

A

Formally, **Strict Order Dominance** - for any two answers A,A' where the usefulness of the i-th document in A, A' is ai, a'i, respectively: If $\exists i > 0 : (\forall j < i : \text{a'i} = \text{aj}) \wedge (\text{a'i} > \text{ai})$

then answer A' is preferable to A.

The DivGen Approach

- ***Our goal – to solve DAS efficiently***
- ***What we are going to talk about ?***

1. A threshold algorithm for DAS – GenFILT algorithm (existing approach)
2. The DivGen algorithm (novel approach)
3. DivGen execution example
4. DivGen algorithm - Access scheduling

A threshold algorithm for DAS - GenFILT(1)

1. **Generate step** - example:

User Query contains feature -“Twitter” with weight=0.5

Twitter: <Blog1, relevance: **0.05=0.5*0.1** >
<Blog2, relevance: **0.1** >

rel(d|q) = sim(q,d)=<q,d>

Blog2 is emitted

	Feature	weight
Doc	**Twitter**	**Obama**
Blog1	0.1	…
Blog2	0.2	…
…	…	…

A threshold algorithm for DAS - GenFILT(2)

- **A Sequential Access (SA)** – on query feature *i*, will retrieve the *id* of the document with the next highest weight for feature *i*. provide the following information: either

i) the exact weight of a feature in a document,

Or

ii) an upper bound on said weight (if the document has not been encountered on any SA on the feature).

inverted index

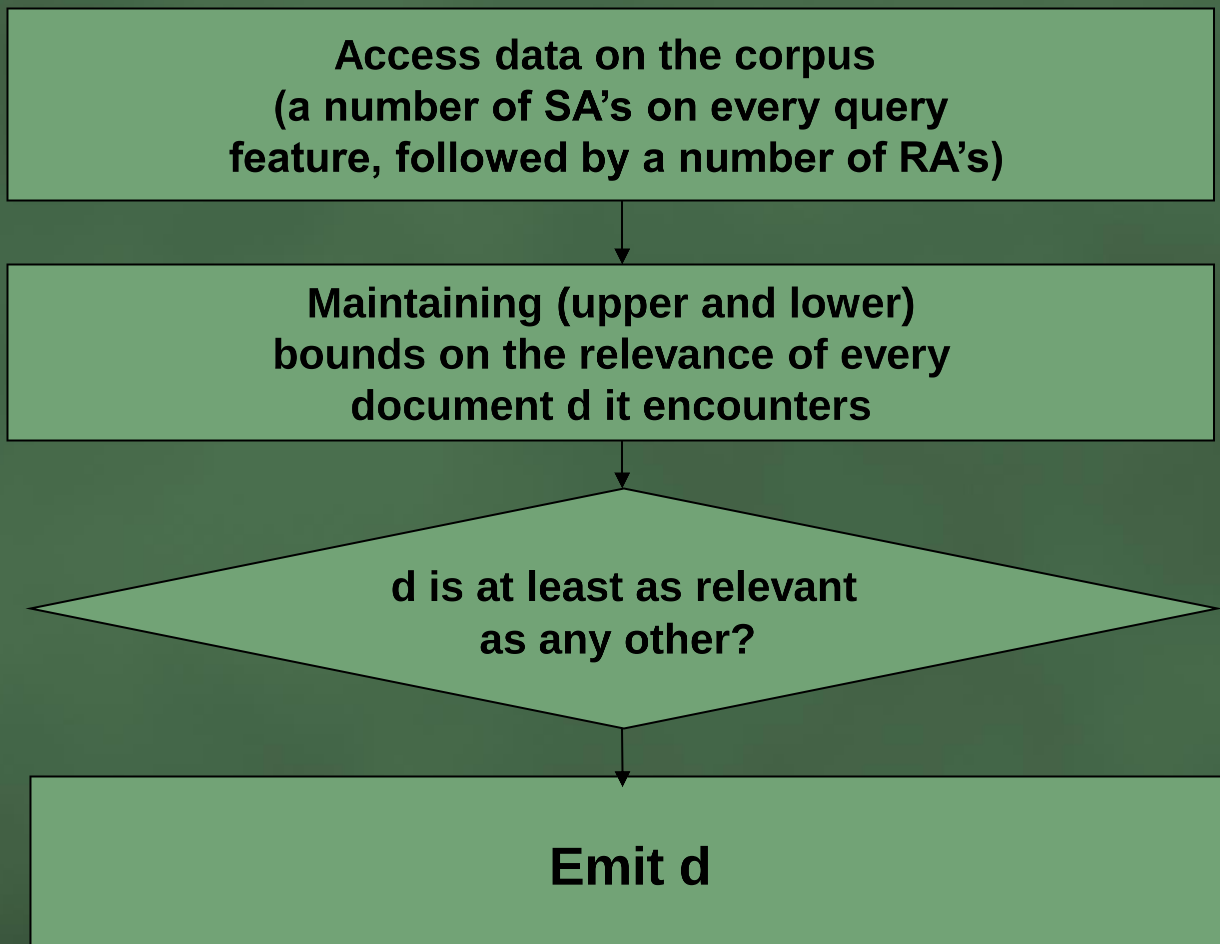
Feature	“Twitter”
Doc	**Weight**
Blog1	0.2
Blog2	0.1
…	…

A threshold algorithm for DAS(3)

- **A Random Access (RA)** - on a feature *i* and document *d* will retrieve the exact weight of *i* in *d* (or 0 if *d* doesn't contain i) – *optionally to improve performance*

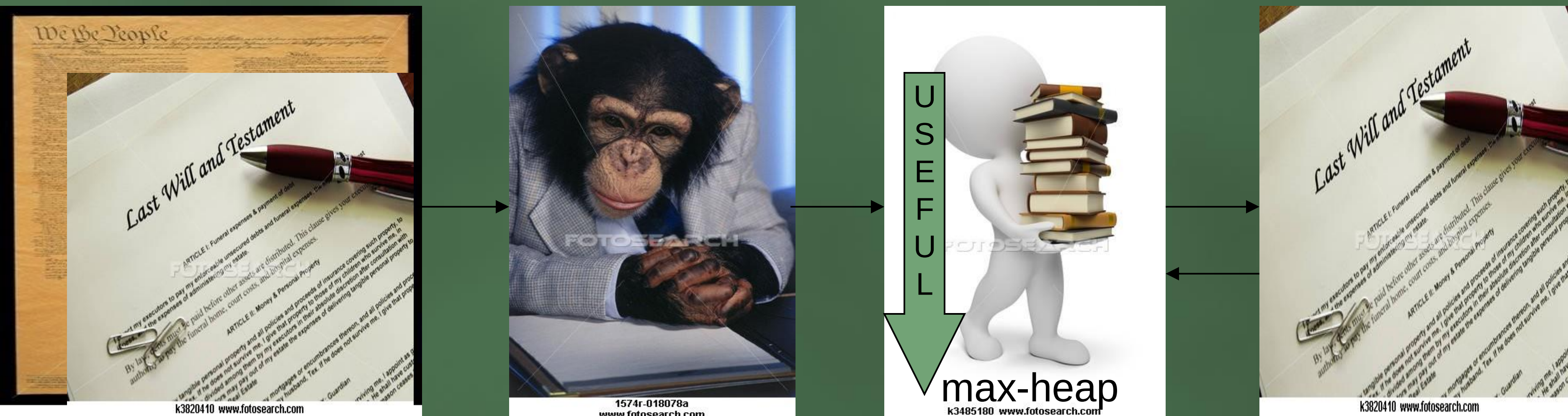
A threshold algorithm for DAS - GenFILT(1)

1. Generate step - outputs documents in descending order of relevance:



A threshold algorithm for DAS(4)

2. Filter step – incrementally reranks them, taking diversity into account:



GENERATE k docs in descending order of relevance

1. Retrieving the actual contents;
2. Compute usefulness wrt. documents already emitted

1. Emit head of max-heap
2. Update usefulness (based on similarity to emitted docs)

GenFILT - drawbacks

- Needs to fully compute the relevance
- Retrieve the entire content, of all top relevant documents, even if they are highly similar to each other (and hence most do not take part in the final answer)
- This results in a lot of wasted I/O effort
- Hardly any early pruning is possible with this approach

The DivGen Algorithm - Novel data access primitives

- **Bound Access (BA)** – on a document *d* will retrieve a features with the highest weight *w* in *d*, as well as an upper bound *w* on the weight on any other features of *d*

- **Batch Sequential Access (BSA)** - on a (non-query) feature *i* will retrieve the documents with the highest weight of *i*, as well as an upper bound *w* on the weight of *i* in any other document

- **Document Random Access (DocRA)** - on a document *d* will retrieve all the features with nonzero weight in *d*, along with their exact weights

The DivGen Algorithm
Data access primitives - summary

Feature	Input	Output
SA	Feature j	Next largest d_i^j
RA	Feature j, document i	d_i^j
BA	Document i	Set of F features {j} with largest d_i^j Min d_i^j among these
BSA	Feature j	Set of F documents {i} with largest d_i^j Min d_i^j among these
DocRA	Document i	All d_i^j

The DivGen Algorithm

1. **Repeat**
2. **Perform** some SA's on every query feature (and any applicable BA's)
3. **Schedule** and perform DocRA's, BSA's and RA's **(we discuss it later)**
4. **"Semi-prune"** candidate documents that have upper bound on usefulness <= the lower bound on usefulness of the current top document
5. **If** only one candidate document remains, with usefulness at least as high as any document not yet encountered then
6. **Emit** current top document
7. **Update** the novelty of all semi-pruned documents, and mark them as candidate documents
8. **Until** min (k, size of corpus) documents have been emitted

DivGen Execution example(1)

- **For this example we assume:**

1. All features have weights of at most *5*, single query feature *x* is used with *w=1*, *k=2* top documents are required

2. For all docs and queries $\sum_i d_i^2 \leq 100$, we set

fq = 1

AND

rel(d|q) = <d,q>/100

AND

red(d|di,q) = <d,di>/100

3. Arbitrary *access scheduling policy* is used, leading to a number of DocRA's, BSA's and/or RA's, after every 2 SA's.

DivGen Execution example(2)

Doc	Weight	Summary
A	5	[x,5]
B	4	[x,4]
C	3	[z,3]
D	2	[x,2]
E	1	[y,2]

(a) Inverted list for x (BA)

	Feature weight		
Doc	x	y	z
A	5	5	1
B	4	4	4
C	3	0	3
D	2	0	1
E	1	2	0
F	0	3	4

(b) Corpus documents (DocRA)

Feature	Summary
x	[A,B,4]
y	[A,B,4]
z	[B,F,4]

(c) Champion lists for BSA's (k=2)

DivGen Execution example(3)

	DocId	Relevance %	Novelty %	Usefulness %	Known Features (reason)
Step 1 (2 SA's on x)	A B others	5=5*1 4=4*1 <=4	100 100 100	5 4 <=4	x = 5(SA), others <= 5(BA) x = 4(SA), others <= 4(BA) x <= 4(SA)
Step 2 (Emit A)	B others	4 <=4	56-80 <=100	2.24 – 3.2 <=4	x = 4(SA), others<= 4(BA) x <= 4(SA)
Step 3 (2 SA's on x)	B C D others	4 3 <=2 <=2	56-80 65-82 78-90 <=100	2.24 – 3.2 1.95 – 2.46 1.56-1.8 <=2	x = 4(SA), others<= 4(BA) x = 3(SA), y<= 3,z => 3(BA) x = 2(SA), others<= 2(BA) x <= 2(SA)
Step 4 (BSA on y for candidate B)	B C	4 3	56-60 65-82	2.24 – 2.4 1.95 - 2.46	x = 4(SA), y = 4(BA+BSA), z <= 4(BA) x = 3(SA), y< = 3,z => 3(BA)
Step 5 (DocRA on C)	B C	4 3	56-60 82	2.24 – 2.4 2.46	x = 4(SA), y = 4(BA+BSA), z <= 4(BA) x = 3(SA), y = 0,z = 3(DocRA)

Access scheduling(line 3 of DivGen algorithm)

- **Our question** - which type of accesses will be performed, on which documents and/or features?
- **The goal** - to perform the data accesses that will lead faster to query processing completion
- **Lets define** the aggregate *uncertainty* of candidate documents - the average difference between their upper and lower bounds of usefulness
- **How to achieve the goal?** – to decrease the *uncertainty*

Access scheduling – *Benefit* definition

- **Lets define** *Benefit* **(at a given point of time)** - the expected reduction in aggregate candidate document uncertainty, if the accesses are performed at that point of time

Access scheduling – *Benefit* estimation(1)

- Given a candidate document *d* we define:

1. usefulness U, its lower bound $\underline{U}$, upper bound $\overline{U}$
2. novelty N, its lower bound $\underline{N}$, upper bound $\overline{N}$
3. feature vector c
4. e_j - unit vector on the j-th dimension (feature)
5. d_i^j - score of the j-th feature of the i-th emitted document (i=1,…,n)
6. documents and queries have $\sum_j (q^j)^2 = 1$
7. Y – number of documents in corpus

Access scheduling – *Benefit* estimation(2)

- The *Benefit* of DocRA:

$$Ben_{DocRA} = \overline{U} - \underline{U}$$

- The *Benefit* of BSA on a non-query feature j, for a single candidate document:

$$Ben_{BSA} = \frac{f_q(\overline{c} - \underline{c})e_j}{2}\left(\sum_j d_i^j\right)\left[\frac{\overline{U}F}{\overline{N}^{1/n}Y} + \frac{\underline{U}(1 - F/Y)}{\underline{N}^{1/n}}\right]$$

- The *Benefit* of RA on a query feature j, for a candidate document:

$$Ben_{RA} = \frac{q^j(\overline{c} - \underline{c})e_j}{2}\left[\overline{N} + \underline{N}\right]$$

Access scheduling – *Benefit* properties

- **DEFINITION 4.1.** Function Ben(D,S,A) quantifies the expected benefit of performing DocRA's on documents in set *D*, BSA's on features in set *S* and RA's on document/feature pairs in set *A*.

  Ben(D,S,A) has the following properties for all D,S,A that are not empty sets and for all *d*, *s*.

  i) Ben(DU{d},S,A)<= Ben(D,S,A)+Ben({d},Ø,Ø)

  ii) Ben(D,SU{s},A)<= Ben(D,S,A)+Ben(Ø,{s},Ø) and

  iii) Ben(D,S,A.{(d,s)})<= Ben(D,S,A)+Ben(Ø,Ø,{(d,s)})

Access scheduling – access cost

- **Cost for every access in processing time** –can be estimated as the average time per access type

- **Our goal** - to solve **Access scheduling problem,** i.e. to perform the accesses with the greatest benefit, having total cost under a given cost budget.

Access scheduling problem - definition

- DEFINITION 4.2:

Select a set D of documents to DocRA on,

a set S of features to BSA on, and

a set A of document/feature pairs to RA on,

s. t.

Cost Budget =>$|D| \cdot$ cost(DocRA)+$|S| \cdot$ cost(BSA)+$|A| \cdot$ cost(RA),

in a way that maximizes **Ben(D,S,A).**

Access scheduling problem ☹☹☹☹☹

- THEOREM 4.3. The access scheduling problem (Def. 4.2) is NP-hard

Proof: non-trivial reduction from densest subgraph, bipartite variant.

[reference – A. Suzuki and T. Tokuyama. Dense subgraph problems with output-density conditions. ACM Trans. Algorithms, 4(4), 2008. or Efficient Diversity-Aware Search full paper]

Intelligent Access Scheduling (greedy approach) - ☺☺☺☺

1. **Compute** the ***Benefit*** of all possible DocRA's, BSA's, RA's on candidate documents in isolation (at a given point of time)
2. **CostSoFar** =0
3. **repeat**
4. **Find** the access A with the highest ***benefit / cost*** ratio
5. **if AccessCost** < **CostBudget** - **CostSoFar** then
6. Perform the access A
7. **CostSoFar** = **CostSoFar** + **AccessCost**
8. **until CostBudget** = **CostSoFar**, or all accesses have been considered

Intelligent Access Scheduling (greedy approach)

- Takes time **linear in the number of candidate accesses** to be performed (given 3 types of access)
- **Theoretical guarantee**:

THEOREM 4.4: In the context of Def. 4.2, consider the following greedy procedure: calculate the expected benefit of each access in isolation, and select the accesses with the best benefit/cost ratio, subject to the available cost budget, B. Let BenG be the sum of actual benefits obtainable by this procedure. Moreover, let Ben* be the maximum sum of actual benefits obtainable overall.

Then $\mathrm{BenG} > \frac{1}{3}\left(1 - \frac{\max(\mathrm{cost}(\mathrm{DocRA}), \mathrm{cost}(\mathrm{BSA}))}{B}\right)(\mathrm{Ben}^*)$

Intelligent Access Scheduling (greedy approach)

COROLLARY 4.5:

Set:

Cost Budget = 10·max(cost(DocRA),cost(BSA)) or greater

Get:

Benefit at least within 30% of the optimal

(i.e., every round of accesses, at least 10 accesses are performed)

Evaluation

- All algorithms implemented in Java 6, using Oracle BerkeleyDB Java Edition, v3.3.74 14
- Ubuntu Linux 8.04
- 1GB of physical memory
- Intel Core2 X6800 CPU clocked at 2.93GHz, utilizing only one processor core
- The execution time of each query- from the moment it was received for processing by our implementation, till the moment results were returned to the user (i.e. excluding the one-time cost of system initialization).
- rel(d|q) = cos(d,q) = <d,q>
- sim(d,di) = cos(d,di)=<d,di>
- Focus parameter - fq = 0.4

Experiments on real data(1)

- Grapevine corpus is taken (onthegrapevine.ca)

- Documents consisted of all 8.6M blog posts made during the month of June 2009 on major blog hosting services

(non-English posts after removal of spam)

- Grapevine had identified over 500K distinct real-world entities (people, locations, products, etc.) in these posts.

Experiments on real data(2)

- The entities found in each document = features
- An average of 3.93 distinct features were found in each document
- Documents and queries are represented as length-normalized vectors
- The weight of each feature:

$$W_{feature} = FeatureFreq_{doc} * \log(FeatureFreq^{-1}{}_{corpus})$$

Experiments on real data(3)

- Two types of queries were used:
  1. "easy" queries - 100 random pairs of popular entities during June 2009
  2. "hard" queries - the set of key entities involved in one of the top 100 engaging stories/events for the month of June 2009; 3.3 entities/query in average

  example for top story in June 2009:

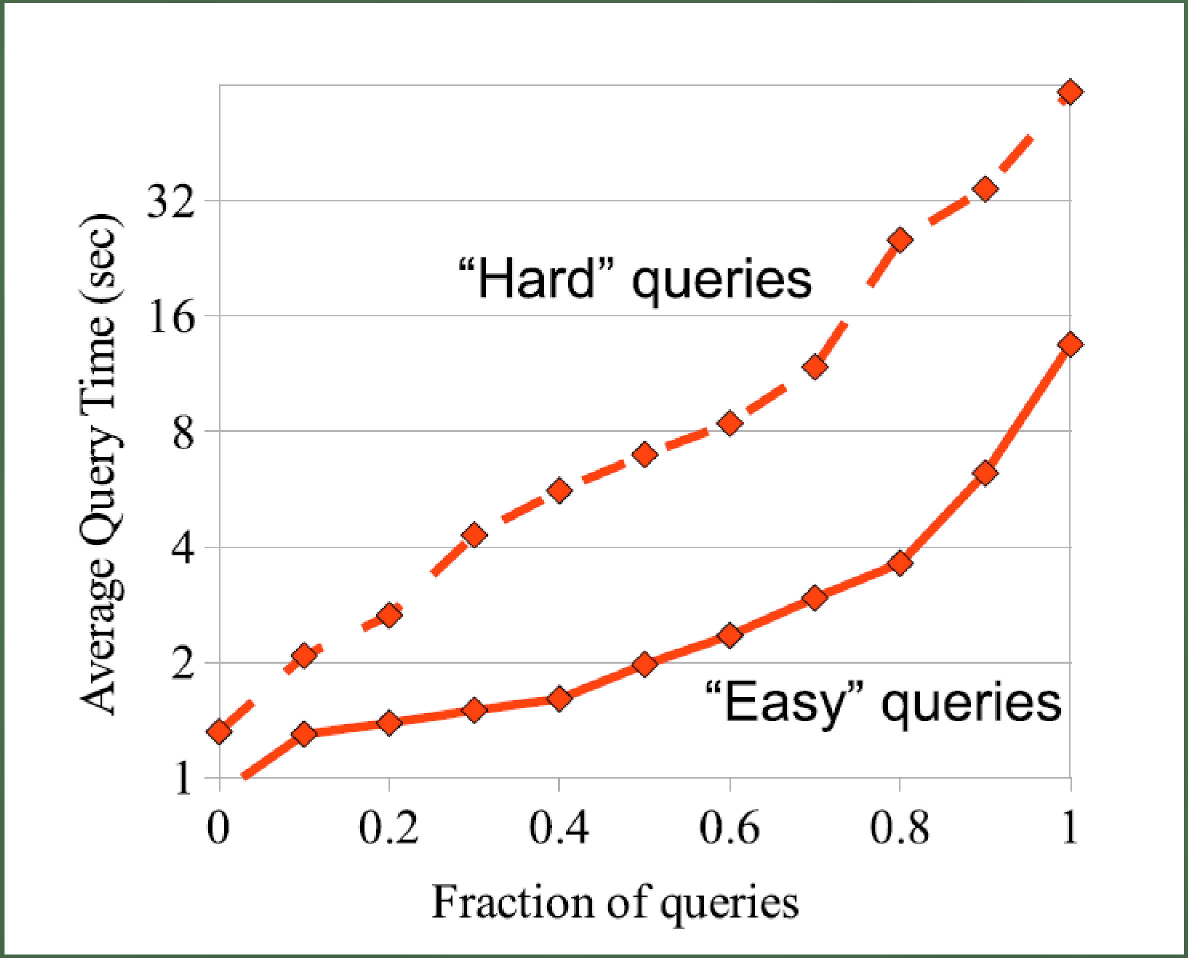
  top story: "Election crisis in Iran",

  entities = {"Iran","A. Ahmadinejad","M.Maussavi"}

- GenFilt and DivGen were executed with k=10(number of top results/query)

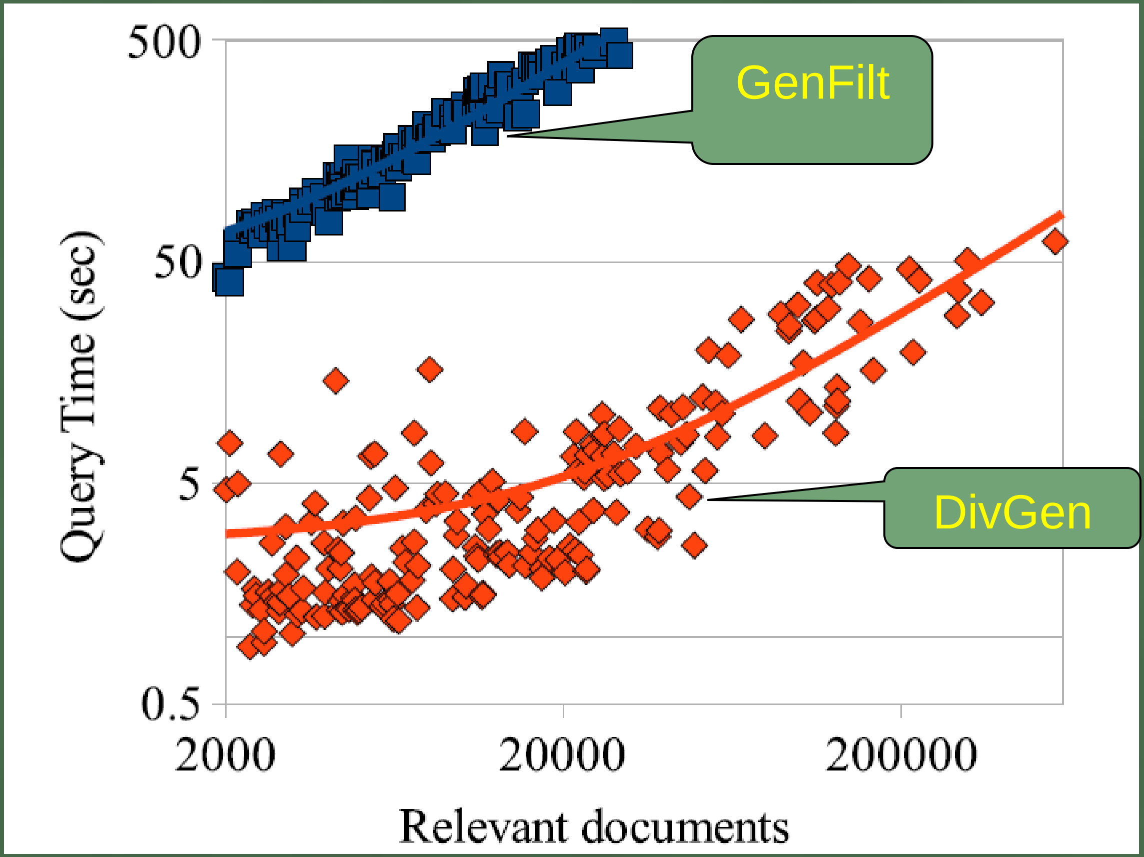
Experiments on real data(4)

- The data point (0.7, 2.9), for the "easy" queryload, signifies that 70% of the "easy"queries terminated in under 2.9 seconds each using **DivGen** .



Experiments on real data(5)

- Scatterplot of query time vs. number of relevant documents, along with linear regression, i.e. documents that contained at least one of the query terms



Experiments on synthetic data – dataset

- Feature weights were assigned as in the real dataset
- The number of feature occurrences in each document followed a normal distribution
- Corpora consisted of 1M documents, with an average of 100 feature occurrences per document (and a relative standard deviation of 10%) and 5K distinct features in the corpus
- Queryload consisted of 500 randomly selected queries
- Each query contained 3 terms
- Requested the top k = 20 results

Experiments on synthetic data- DivGen data access scheduling

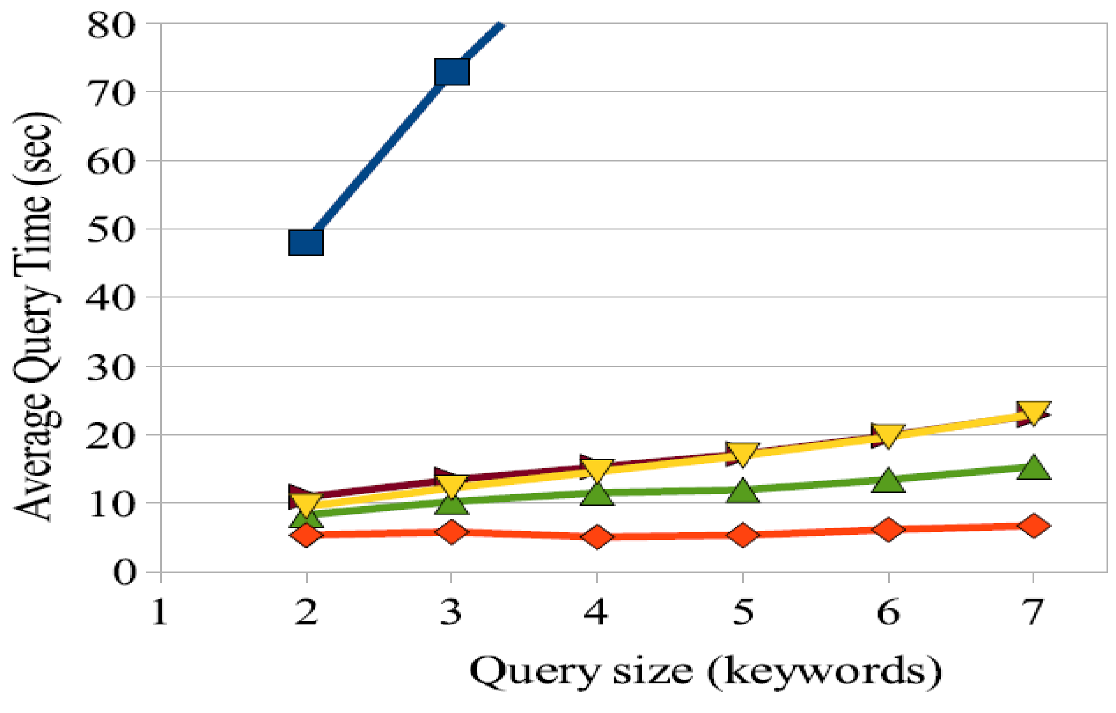
- 4 variations of DivGen were used:
  1. Original DivGen implementation
  2. DivGen- BSA – used a fixed number of BSA's
  3. DivGen- RA – used a fixed number of RA's
  4. DivGen- All – used a fixed number of BSA's and RA's

Note, all three last variants performed a fixed number of DocRA's and the same cost budget

Experiments on synthetic data - results

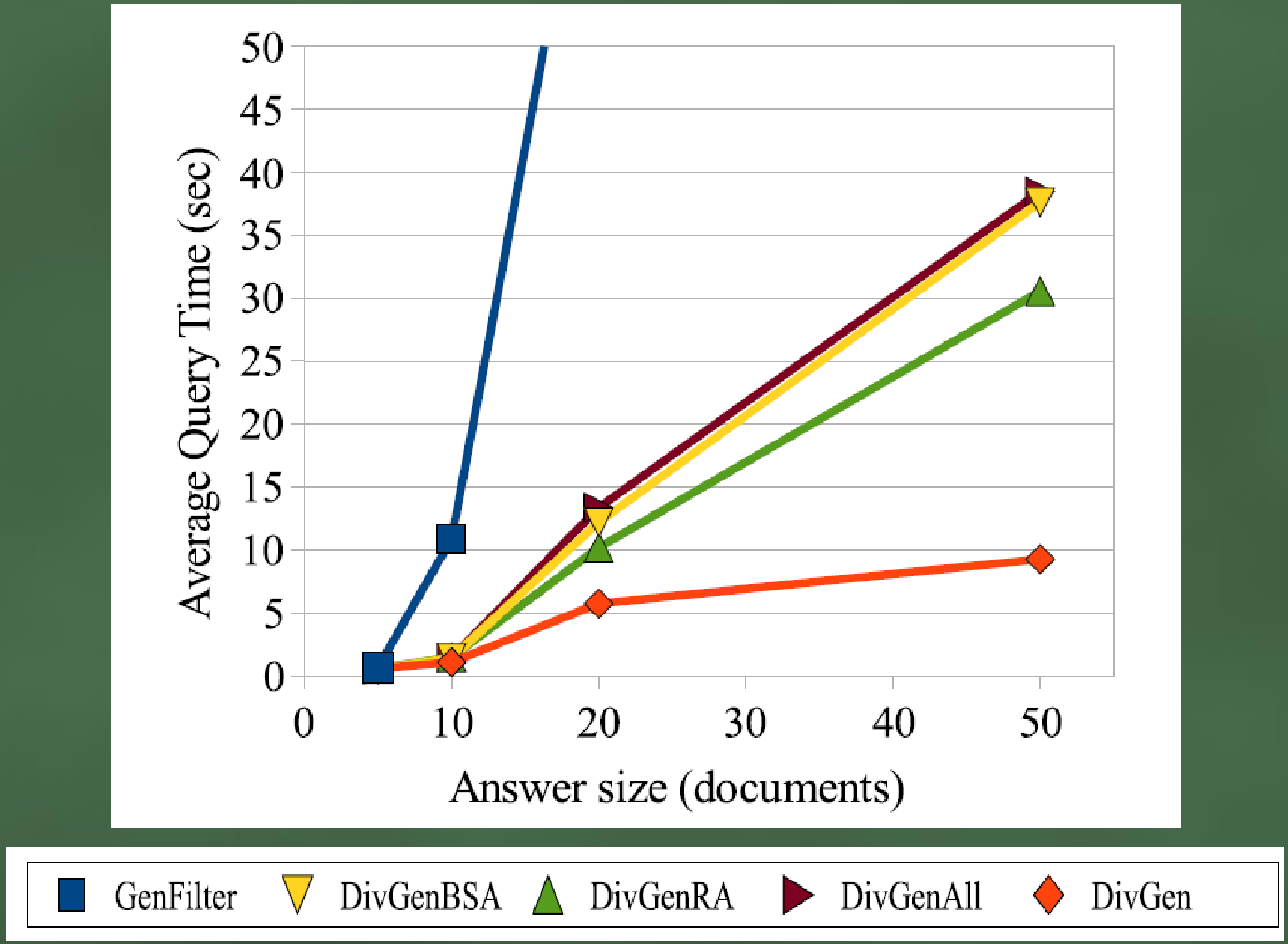
- All algorithms (GenFilt and DivGen, and its 3 variants) were executed for :
  1. Parameters related to the ***size*** of the problem (query size, answer size)
  2. Parameters related to the ***difficulty*** of the problem (query focus parameter, average document size, etc.)

Experiments on synthetic data – results, query size



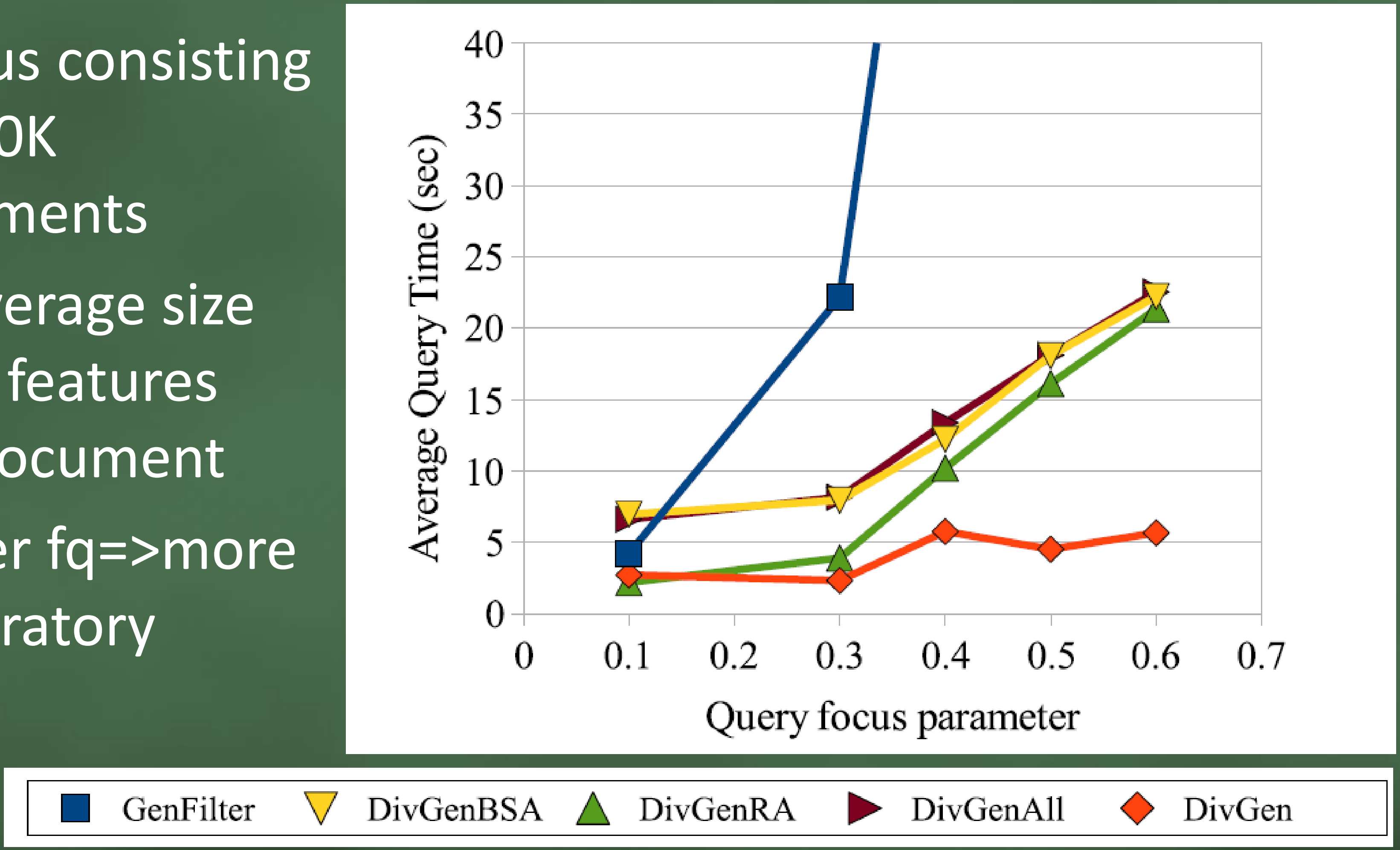
(c) Query size

Experiments on synthetic data – results, answer size



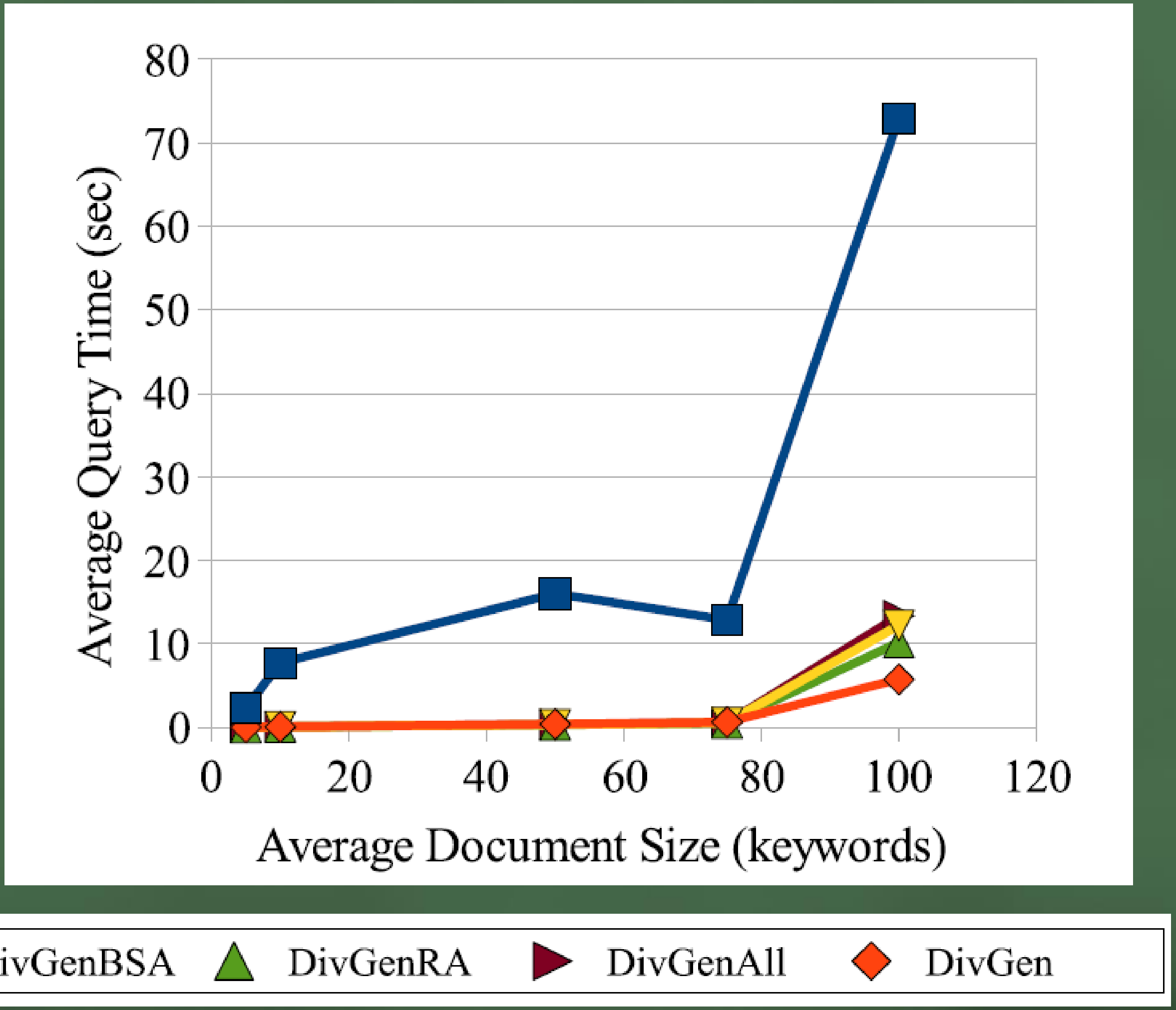
Experiments on synthetic data – results, query focus parameter

- Corpus consisting of 500K documents
- An average size of 50 features per document
- Higher fq=>more exploratory



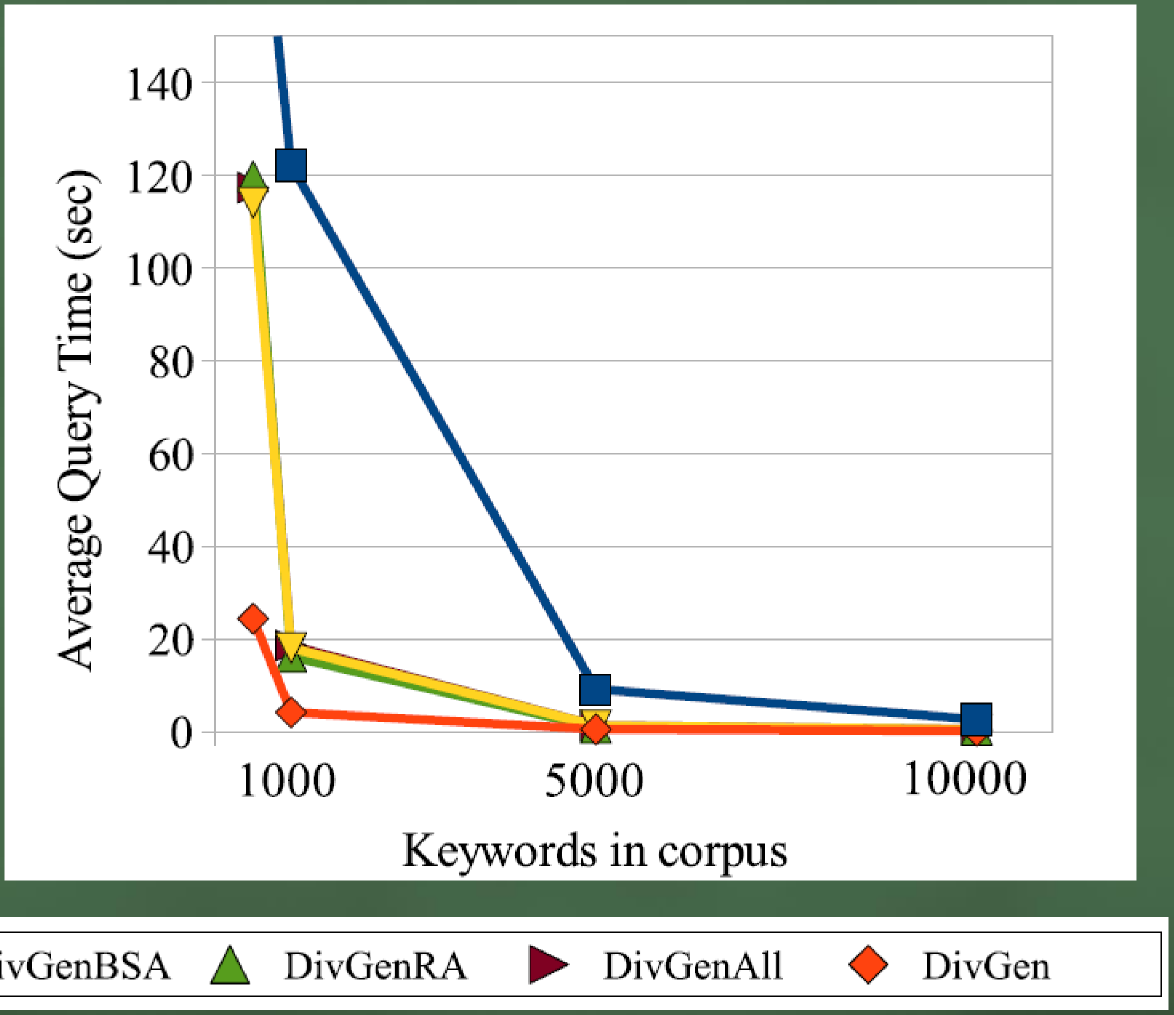
Experiments on synthetic data – results, average document size

- The average number of feature occurrences (keywords) in a document is varied
- Fixed number of distinct features in the corpus is used



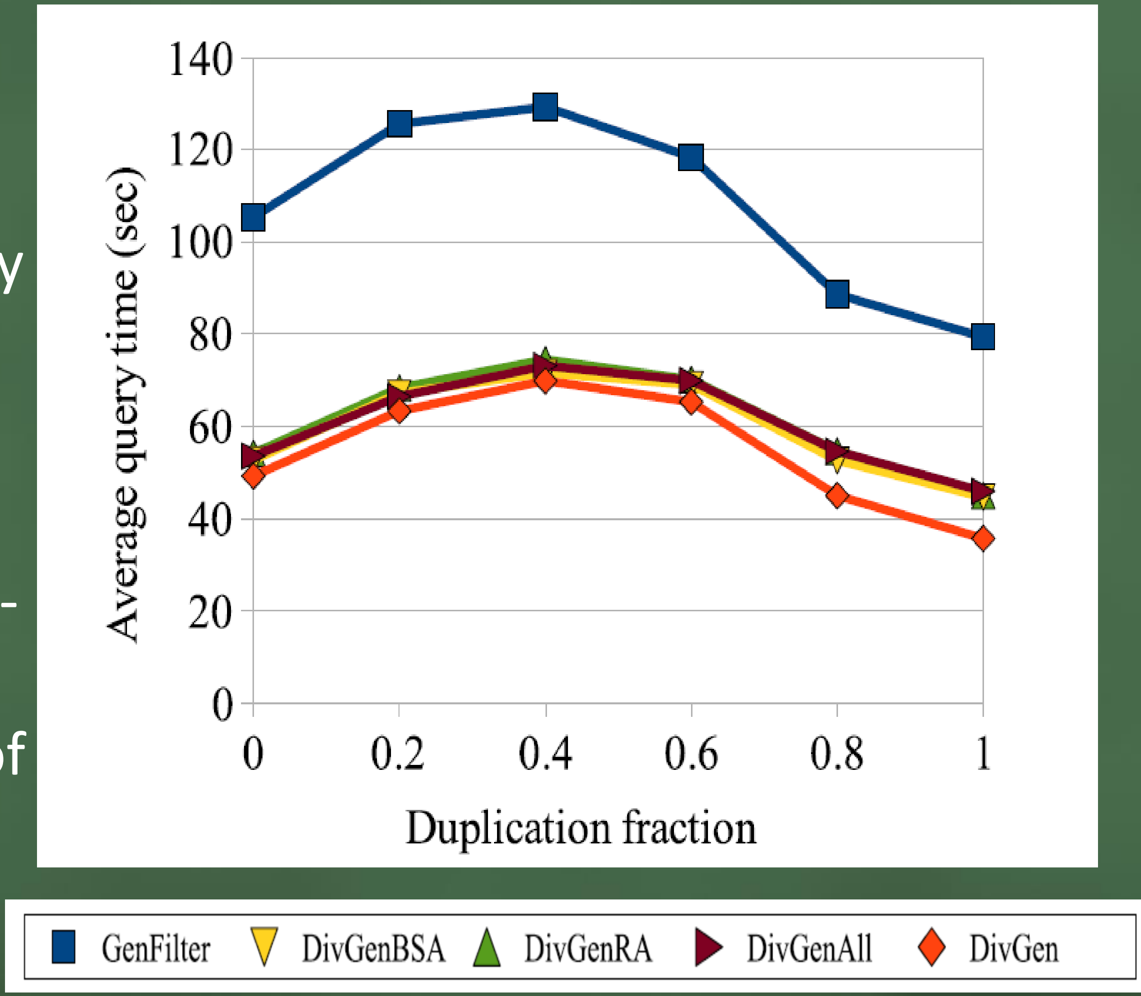
Experiments on synthetic data – results, dictionary size

- The number of distinct features in the corpus is varied
- A smaller number of distinct features will result in a higher similarity between documents
- Corpus consisting of 500K documents



Experiments on synthetic data – results, correlated dataset

- Corpus size = 500K of documents
- Group documents in 100 equal size groups arbitrarily
- Select an arbitrary document in every group
- Copy 90% of its content over to a fixed number (1K-5K) of documents in its group, while erasing 90% of the original content of these documents



Experiments on synthetic data – I/O behaviour

- On average, 80% of the processing time for each query was due to I/O, for all algorithms
- In 80% of the queries, DivGen variants spent 70%-90% of processing time on I/O
- In 80% of the queries GenFilt spent 80%-90% of processing time on I/O

Effectiveness - experimental setup

- Real dataset is used
- Examined two **news search tasks**
  1. *highly* exploratory – “current news”
  2. *moderately* exploratory nature – “news about…”
- Compare “top k, then rerank” heuristic(MMR) and DivGen
- 96 human evaluators (after spam removal), were asked to compare results produced by pairs of different conditions, for 5 tasks of type “Current news” and 5 of type “News about · · · ”

Effectiveness –"Current news"

- Identifying news stories, across all domains of interest, that captured popular attention in a given period, as evidenced by the social media collective
- For some days in June 2009, used as queries the set of top entities that were discussed by people on social media each day, as identified by Grapevine
- For each day, retrieved the top-5 blog posts made during the preceding 4 day period, for varying query focus parameters

Effectiveness –"Current news"

Id	Snippet	Id	Snippet	Id	Snippet
	No diversification (f_q=0)		DivGen : Moderate diversification (f_q=0.4)		DivGen : High diversification (f_q=0.7)
41	President Obama[...],Health Care & Stimulus Plans	41	President Obama[...],Health Care & Stimulus Plans	41	President Obama[...],Health Care & Stimulus Plans
42	Obama's Speech in Cairo	46	No one talking about dumping dollar:China minister	50	Trying to Put the 'O' Back in Orlando
43	Obama Submits [Speech in Cairo]	47	[Reaction to] Orlando Magic [first NBA final victory]	51	Open Letter to Microsoft: [...]Mobile Strategy
45	"Obama's Cairo Speech"	81	Apple [keynote, announcing cheaper] iPhone 3GS	83	The Taliban bites back
80	Text of Obama's Cairo Speech	82	New York Yankees take on Boston Red Sox	84	Gameday Live:Yankees at Red Sox
	MMR (θ = 10, f_q=0.4)		MMR (θ = 50, f_q=0.4)		MMR (θ = 100, f_q=0.4)
41	President Obama[...],Health Care & Stimulus Plans	41	President Obama[...],Health Care & Stimulus Plans	41	President Obama[...],Health Care & Stimulus Plans
42	Obama's Speech in Cairo	46	No one talking about dumping dollar:China minister	46	No one talking about dumping dollar:China minister
43	Obama Submits [Speech in Cairo]	86	Top Ten Myths about the Middle East	86	Top Ten Myths about the Middle East
85	Remarks of President Barack Obama	87	Can Obama reconcile[...] health care reform	87	Can Obama reconcile[...] health care reform
45	"Obama's Cairo Speech"	88	To President Obama Re: Islam and Science	89	Protests against Putin sweep Russia

(a) News for June 10th, 2009 (107 entities)

Effectiveness –"Current news", conclusion

- 2/3 of human evaluators rated diversified results as better than non-diversified
- The results for MMR were almost identical for high diversification fq = 0.4
- 62% of participants rated DivGen as better than MMR

Effectiveness – ”News about…”

- Using as queries some topics that were popular during June 2009 (as identified by Grapevine)
- Retrieved the top-5 blog posts for each, for varying query focus parameters
- Report their urls and a brief description of their contents for one such topic, “Mahmoud Ahmadinejad”

Effectiveness –" News about…"

No diversification (f_q=0)

1	Ahmedinejad is getting a run for his money
2	Will Iran's 'Marriage Crisis' Bring Down Ahmadinejad?
3	Iranian presidential debates
4	Why Ahmadinejad won Iran's election
5	Ahmadinejad supporters and some of their actions

DIVGEN : Moderate diversification (f_q=0.4)

1	Ahmedinejad is getting a run for his money
2	Will Iran's 'Marriage Crisis' Bring Down Ahmadinejad?
6	Iran's Green Wave:Ahmadinejad's Undoing?
7	Neither Ahmadinejad nor Mousavi
8	Iranian Clerics Take To The Streets

DIVGEN : High diversification (f_q=0.7)

1	Ahmedinejad is getting a run for his money
9	Protesting an election
10	That poll[...]showing 2-1 Ahmadinejad support
11	Mousavi Might Not Be Much Better, But...
12	Iran's Disputed Election

(b) News about 'M.Ahmadinejad'

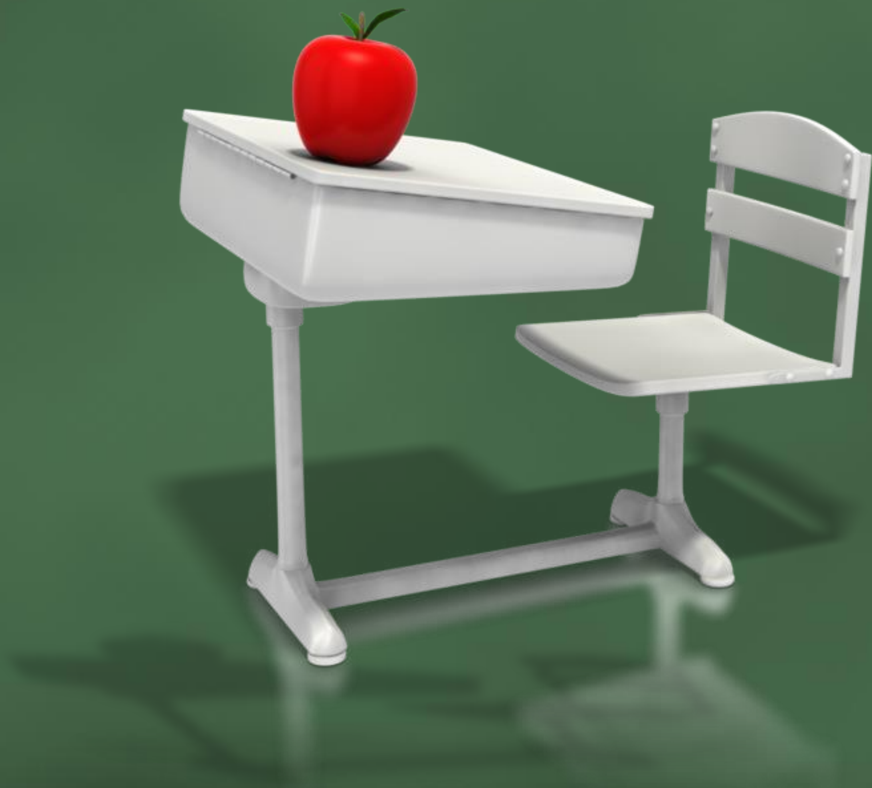
Effectiveness – ”News about…” , conclusion

- Diversity awareness does not significantly boost the answer quality
- 58% of evaluators rated diversified results as better than non-diversified

Conclusions

- DivGen an efficient threshold algorithm for diversity-aware search was presented
- DivGen utilizes novel data access primitives, offering the potential for significant performance benefits
- Proposed a low-overhead, intelligent data access prioritization scheme, with theoretical quality guarantees, and good performance in practice
- The efficiency and effectiveness of our approach with a comprehensive experimental evaluation were validated

Questions???



Thank you