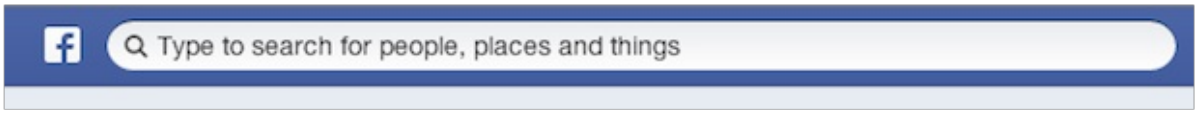
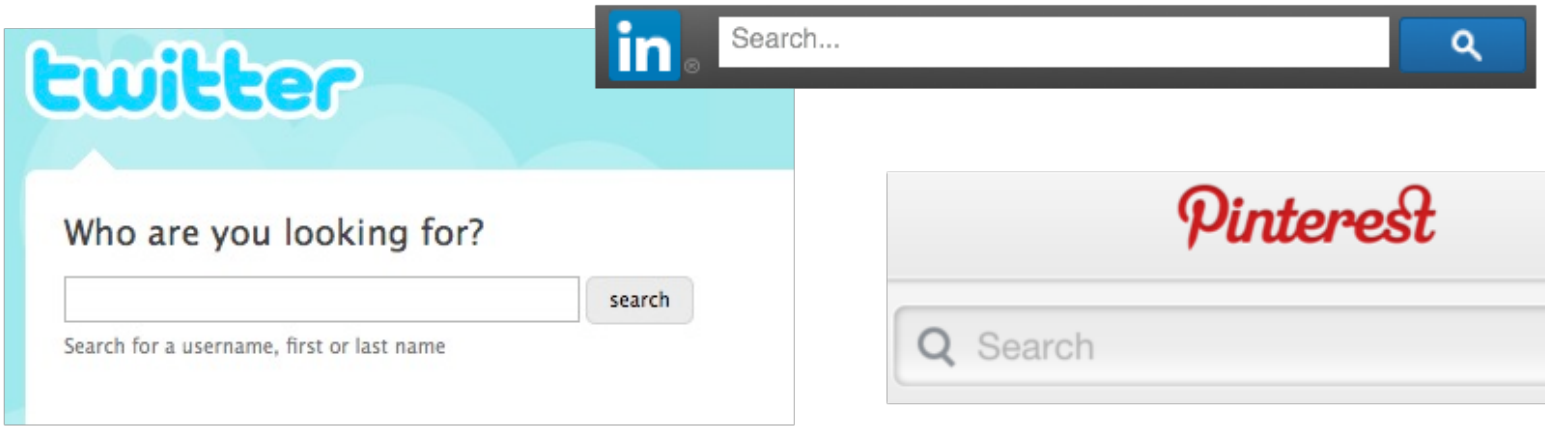
Toward Social, Structured and Semantic Search

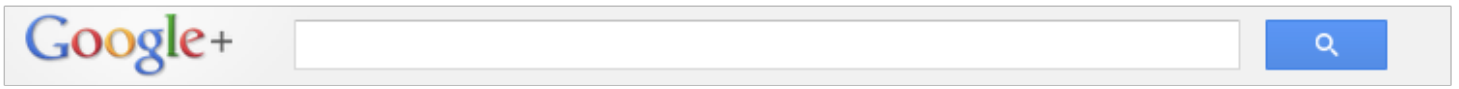
Raphaël Bonaque[1,2], Bogdan Cautis[2,1], François Goasdoué[3,1], Ioana Manolescu[1,2]

[1] INRIA [2] Université Paris-Sud [3] Université de Rennes 1

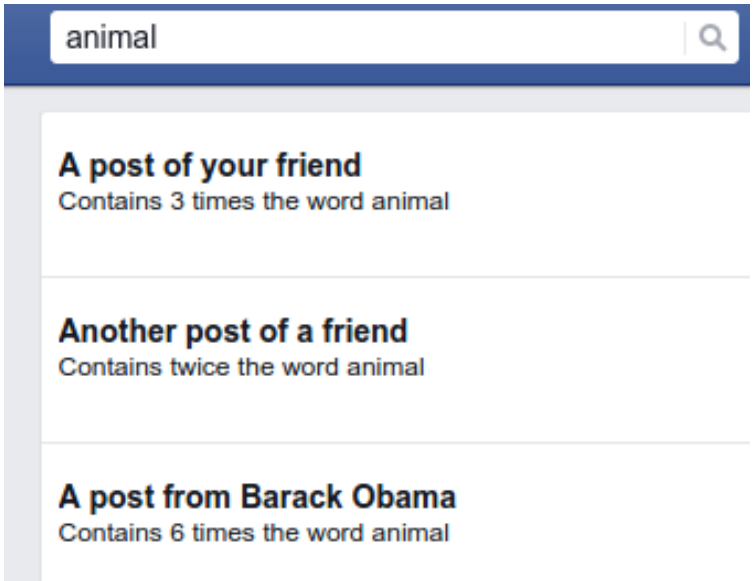
Social network: from the search perspective





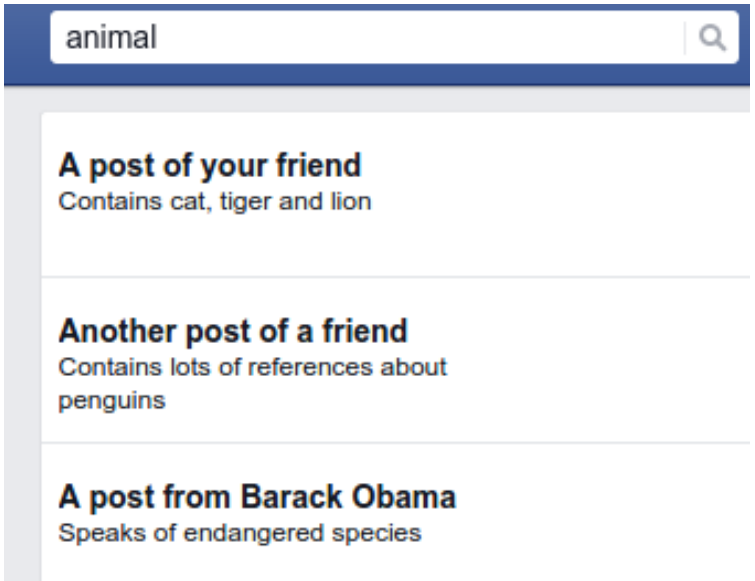


What do you get ?



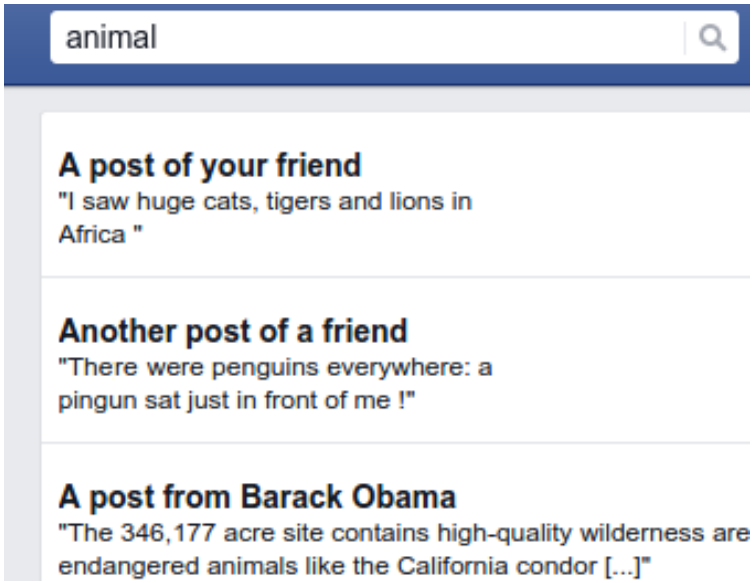
- personalised results
- that contain your keywords

What do you want ?



- personalised results
- that contain **the meaning** of your keywords

What do you want ?



- personalised results
- that contain **the meaning** of your keywords
- **the relevant part of the results**

Toward

- Social: personalised results based on the social network
- Semantic: integrate the semantic of the keywords
- Structure: find most pertinent extracts of arbitrary long documents

(• Top-k Search: only the k best results)

Guidelines

1. Unified model
   - with structure
   - . . . social
   - . . . semantic
2. Query model
   - Keyword extension
   - Social distance with structure
   - Top-k
3. Implementation and evaluation
   - Implementation
   - Evaluation
4. Future and related works
   - Futures work and conclusion

1

Unified model

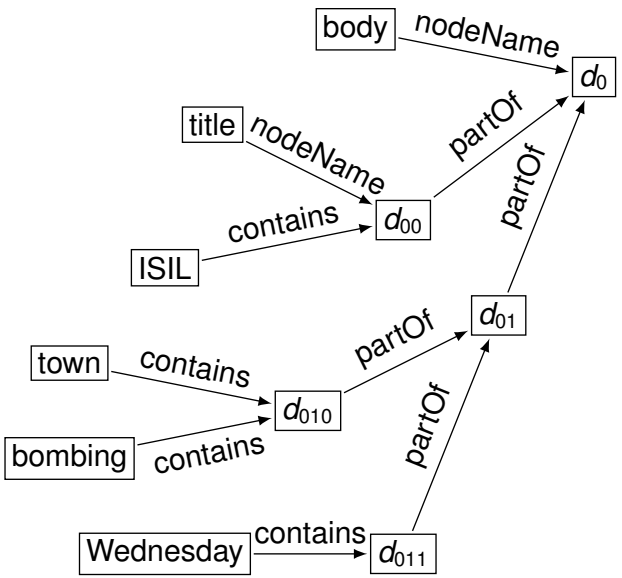
A unified model

title: News on ISIL
text:
- The bombing over the town of ....
...
- On Wednesday ...

Models most popular web formats: XML and JSON

- Documents are organized as *ordered unranked trees*
- Every node has *a name: a single keyword or none*
- Every node has *content: a bag of keywords*

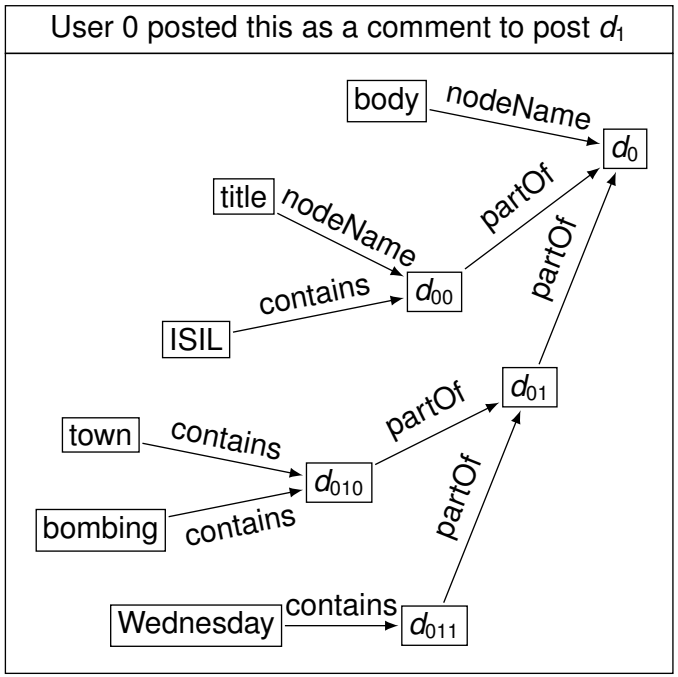
A unified model

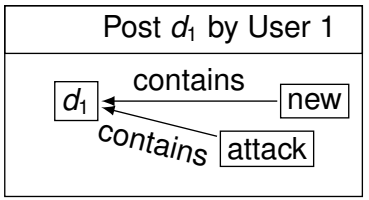


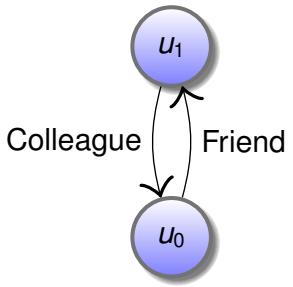
Models most popular web formats: XML and JSON

- Documents are organized as *ordered unranked trees*
- Every node has *a name: a single keyword or none*
- Every node has *content: a bag of keywords*

A unified model



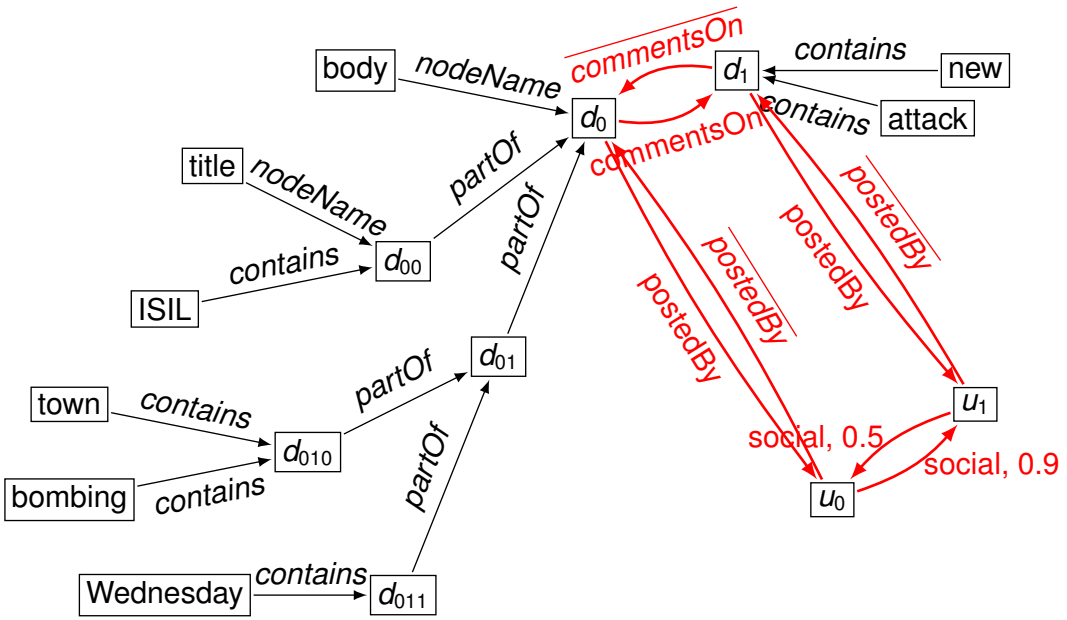




Models most social networks and platforms: Facebook, Twitter, Google+, Tumblr, . . .

- Weighted interactions between users
- Document might comment/answer each other
- Tags, with keywords, are possible on documents
- Authorship for documents and tags

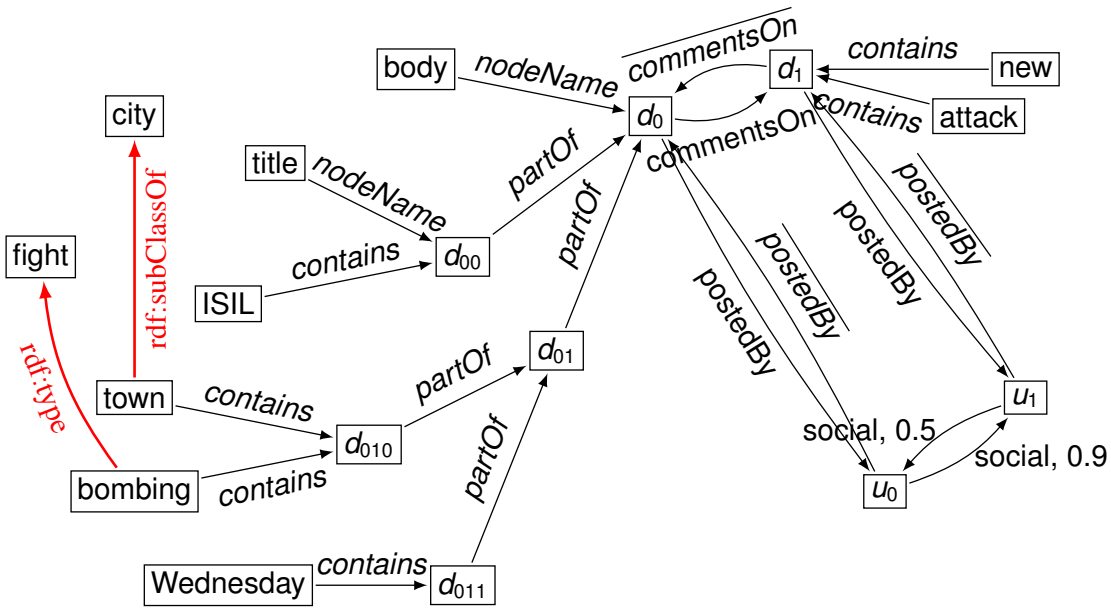
A unified model



Models most social networks and platforms: Facebook, Twitter, Google+, Tumblr, . . .

- Weighted interactions between users
- Document might comment/answer each other
- Tags, with keywords, are possible on documents
- Authorship for documents and tags

A unified model



RDF

Unit : RDF triple

subject property object

E.g., cat likes milk

subject, property and *object are keywords:* ***what we search on***

RDFS allows expressing data semantics by means of rules, hence inferred data.

E.g.,

cat likes milk

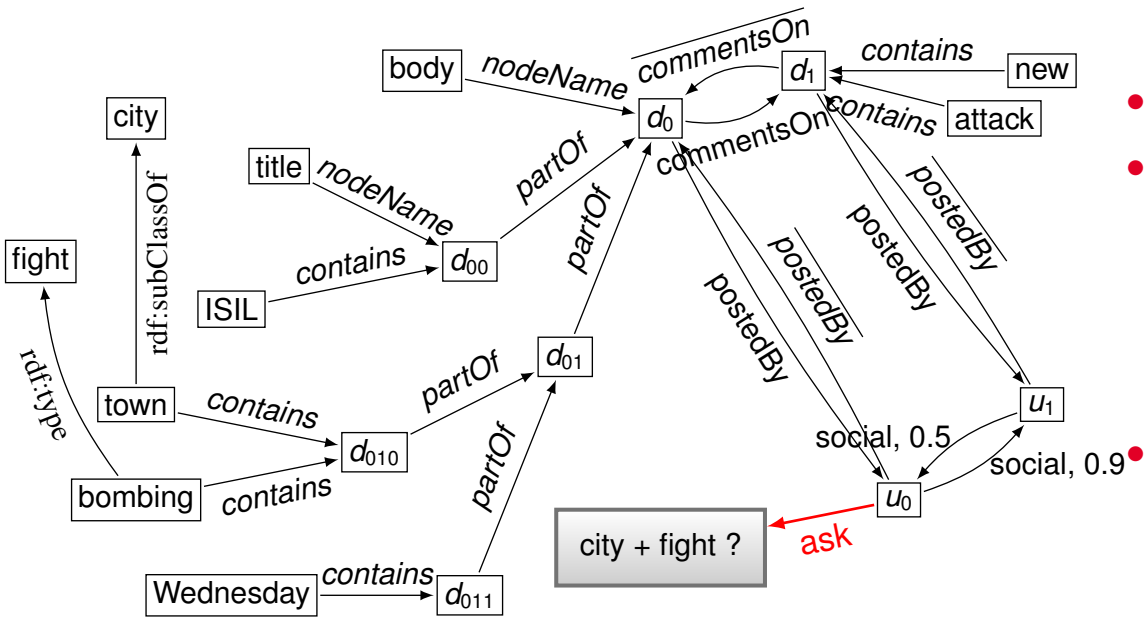
$\wedge$ likes rdfs:domain sentient_being

$\vdash_{\text{RDF}}$ cat rdf:type sentient_being

2

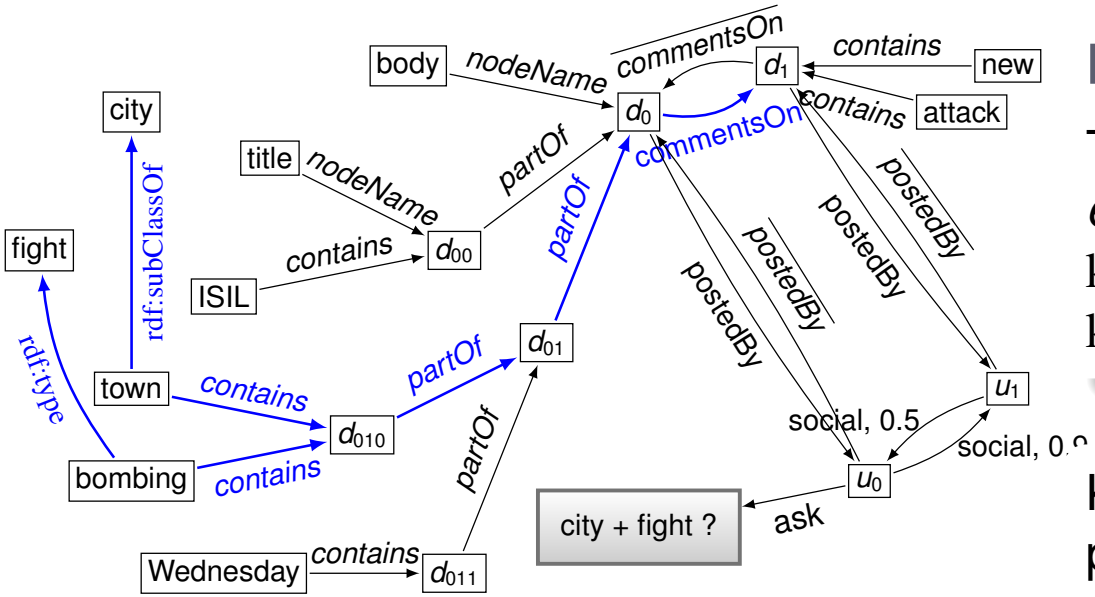
Query model

Query



- *A (user, keyword set) pair*
- Query answer: k documents (or document subtrees) in score order
  - Such that none is a subtree of another
- Score reflects: relevance to tags; social and document distance

Query

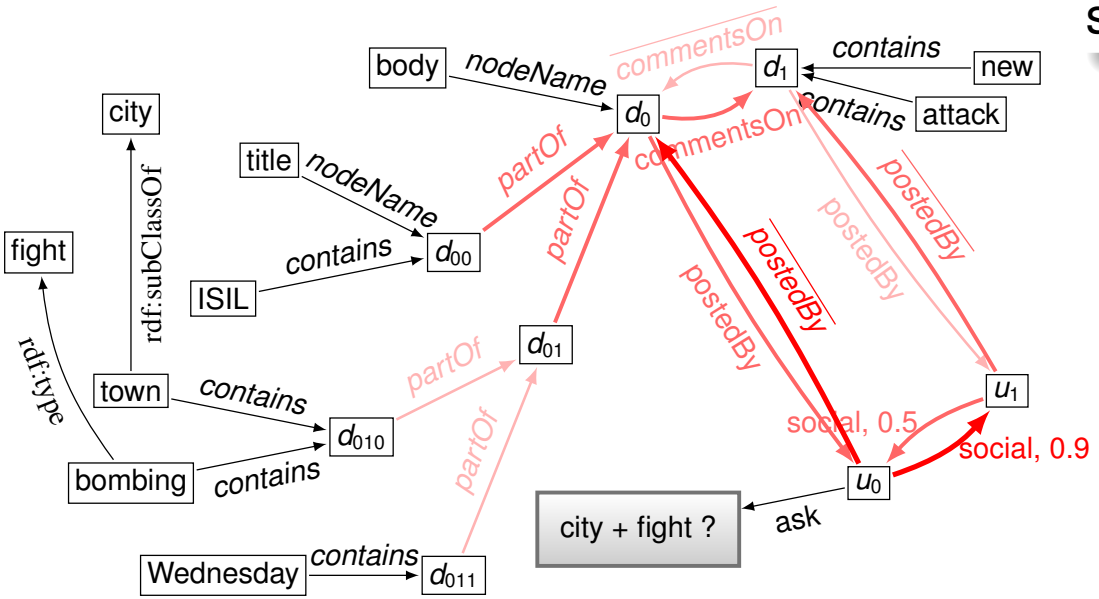


Keyword extension

The relevant keywords:
$ext(k) = \{k' \mid \text{k}' \text{ rdf:type k, or } \text{k}' \text{ rdfs:subClassOf k or } \text{k}' \text{ rdfs:subPropertyOf k}\}$

Keyword meaning is propagated up structure and comments.

Query



Classical distances on the social part of the graph, with a twist to accommodate for structure.

- Social component of the score: aggregation over social paths
- Social path: chain of edges from the social network **plus jumps between ancestors or descendants within documents** : two persons commenting on different part of a document don't make them closer

Query

The semantic and social + structural distances are aggregated into a score, then iteratively:

- we select the best document
- we now ignore its ancestors and descendants

Until we have k documents

- Aggregation functions are generic to adapt to your use case
- **In practice we don't compute the score for every document** but via step by step exploration that stops as soon as possible

3

Implementation and evaluation

Implementation

- Python + PostgreSQL implementation
- completely modular, around 3k lines of code
- a rich, customisable distance function that uses every path from the querier to the results
- some non trivial optimisations: early stopping, encoding the instance in memory as much as possible, fast approximation of candidates bounds . . .

Dataset and evaluation

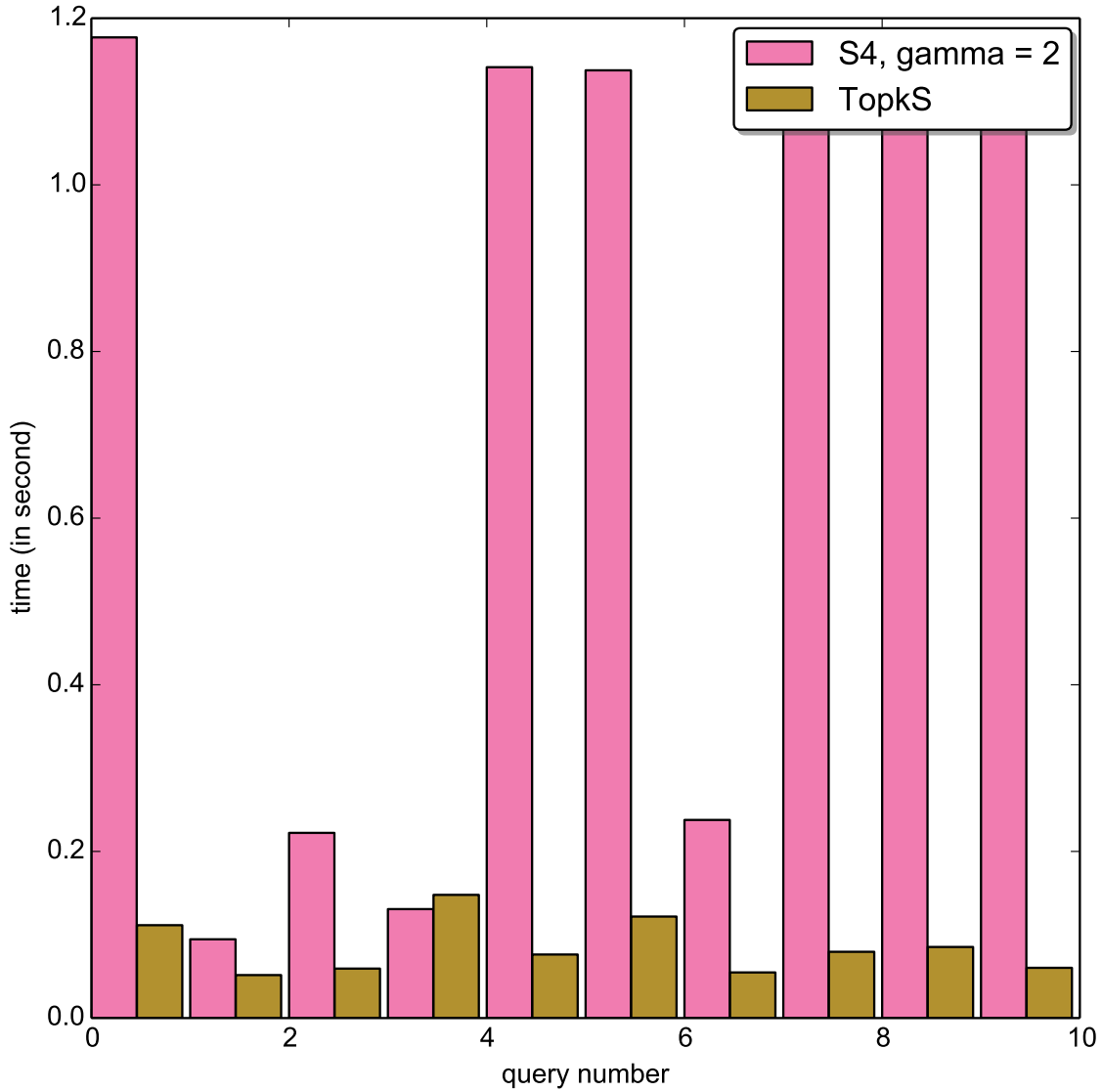
- social content extracted from Twitter: 1 million tweets 0.5 million users
- keyword extension from DBpedia: 28 millions keywords, 160 millions extensions
- relation between users generated from the similarity of their content: 18 millions links
- one big component (120 thousands nodes) and a lot of small ones (< 10 nodes)

Dataset and testing

Comparison with another system: TOPKS [S. Maniu and B. Cautis. Network-aware search in social tagging applications: instance optimality versus efficiency. In CIKM, 2013] :

- Social tagging application : user-item-tag relations and user-user relations
- Shortest distance on the network
- No semantics

Comparison with other system



Comparison with other system

- We are slower **but** we do much more:
  - TOPKS cannot find 94 % of the results we find because it uses only simple path
  - TOPKS ignores semantics and would miss 17% of the results because of that
  - TOPKS ignores the complexity of the network and uses only the shortest path
- Decent runtimes still around 1 second . . . and we are still optimizing

4

Future and related works

- Current work :
  - Improve our algorithm performances
- Future works:
  - Find more interesting data to work on: it is easy to get lot of data but it is hard to get complete data, possible tracks:
    - Vodkaster: a French social network on cinema
    - Tumblr via a partnership with Yahoo!
  - Create new means to evaluate the results

Related Works

The current related work only use restricted dimensions:

- Only structure [L. J. Chen and Y. Papakonstantinou. Supporting top-k keyword search in XML databases. In ICDE. IEEE, 2010]
- Only social [R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, and G. Weikum. Efficient top-k querying over social-tagging networks. In SIGIR, 2008], [S. A. Yahia, M. Benedikt, L. V. Lakshmanan, and J. Stoyanovich. Efficient network aware search in collaborative tagging sites. PVLDB, 2008], [S. Maniu and B. Cautis. Network-aware search in social tagging applications: instance optimality versus efficiency. In CIKM, 2013]
- Only semantic [M. Theobald, R. Schenkel, and G. Weikum. Efficient and self-tuning incremental query expansion for top-k query processing. In SIGIR, 2005]
- Semantic and structure [F. Goasdoué, K. Karanasos, Y. Katsis, J. Leblay, I. Manolescu, and S. Zampetakis. Growing triples on trees: an XML-RDF hybrid model for annotated documents. VLDB Journal, 2013]

Conclusion

- We have a unified model S4: Social, Strucured and Semantic top-k Search
- We implemented and proved a baseline algorithm for top-*k* on S4 data
- More testing on real life datasets are on the way

Questions ?

Thank you for listening !