Semistructured Data Search

Krisztian Balog
University of Stavanger
The data landscape

- **Semistructured data**
  - Lack of fixed, rigid schema
  - No separation between the data and the schema, self-describing structure (tags or other markers)
Motivation

- Supporting users who cannot express their need in structured query languages
  - SQL, SPARQL, Inquery, etc.

- Dealing with heterogeneity
  - Users are unaware of the schema of the data
  - No single schema to the data
Semistructured data

- **Advantages**
  - The data is not constrained by a fixed schema
  - Flexible (the schema can easily be changed)
  - Portable
  - Possible to view structured data as semistructured

- **Disadvantages**
  - Queries are less efficient than in a constrained structure
In this talk

- How to **exploit the structure available in the data** for retrieval purposes?
- Different types of structure
  - Document, query, context
- Working in a Language Modeling setting
- Number of different tasks
  - Retrieving entire documents
    - I.e., no element-level retrieval
  - Textual document representation is readily available
    - No aggregation over multiple documents/sources
Incorporating structure
Preliminaries

Language modeling
Language Modeling

- Rank documents $d$ according to their likelihood of being relevant given a query $q$: $P(d|q)$

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)} \propto P(q|d)P(d)$$

**Query likelihood**
Probability that query $q$ was “produced” by document $d$

**Document prior**
Probability of the document being relevant to any query

$$P(q|d) = \prod_{t \in q} P(t|\theta_d)^{n(t,q)}$$
**Language Modeling**

**Query likelihood scoring**

\[ P(q|d) = \prod_{t \in q} P(t|\theta_d)^{n(t,q)} \]

Number of times \( t \) appears in \( q \)

- **Document language model**
  - Multinomial probability distribution over the vocabulary of terms
  \[ P(t|\theta_d) = (1 - \lambda)\frac{n(t,d)}{|d|} + \lambda P(t|C) \]
  - **Empirical document model**
    - Maximum likelihood estimates
    \[ \frac{n(t,d)}{|d|} \]
  - **Collection model**
    - \( \sum_d \frac{n(t,d)}{|d|} \)
Language Modeling

Estimate a multinomial probability distribution from the text

Smooth the distribution with one estimated from the entire collection

\[ P(t|\theta_d) = (1 - \lambda)P(t|d) + \lambda P(t|C) \]
In the town where I was born,
Lived a man who sailed to sea,
And he told us of his life,
In the land of submarines,

So we sailed on to the sun,
Till we found the sea green,
And we lived beneath the waves,
In our yellow submarine,

We all live in yellow submarine,
yellow submarine, yellow submarine,
We all live in yellow submarine,
yellow submarine, yellow submarine.
Empirical document LM

\[ P(t|d) = \frac{n(t, d)}{|d|} \]
Alternatively...

submarine

yellow
Scoring a query

\[ q = \{ \text{sea, submarine} \} \]

\[ P(q|d) = P(“\text{sea}”|\theta_d) \cdot P(“\text{submarine}”|\theta_d) \]
Scoring a query

\[ q = \{ \text{sea, submarine} \} \]

\[ P(q|d) = P(\text{“sea”} | \theta_d) \cdot P(\text{“submarine”} | \theta_d) \]

\[ (1 - \lambda)P(\text{“sea”} | d) + \lambda P(\text{“sea”} | C) \]

| t       | P(t|d) |
|---------|--------|
| submarine | 0.14   |
| sea      | 0.04   |
| ...      |        |

| t       | P(t|C)   |
|---------|----------|
| submarine | 0.0001  |
| sea      | 0.0002   |
| ...      |          |
Scoring a query

\[ q = \{ \text{sea, submarine} \} \]

\[
P(q|d) = P(\text{“sea”} | \theta_d) \cdot P(\text{“submarine”} | \theta_d)
\]

\[
(1 - \lambda)P(\text{“submarine”} | d) + \lambda P(\text{“submarine”} | C)
\]

| t       | P(t|d) |
|---------|--------|
| submarine | 0.14   |
| sea      | 0.04   |
| ...      |        |

| t       | P(t|C) |
|---------|--------|
| submarine | 0.0001 |
| sea      | 0.0002 |
| ...      |        |
Part I
Document structure
In this part

- Incorporate document structure into the document language model
  - Represented as *document fields*

\[
P(d|q) \propto P(q|d)P(d) = P(d) \prod_{t \in q} P(t|\theta_d)^{n(t,q)}
\]

Document language model
Use case
Web document retrieval
Web document retrieval
Unstructured representation

PROMISE Winter School 2013
Bridging between Information Retrieval and Databases
Bressanone, Italy 4 - 8 February 2013
The aim of the PROMISE Winter School 2013 on "Bridging between Information Retrieval and Databases" is to give participants a grounding in the core topics that constitute the multidisciplinary area of information access and retrieval to unstructured, semistructured, and structured information. The school is a week-long event consisting of guest lectures from invited speakers who are recognized experts in the field. The school is intended for PhD students, Masters students or senior researchers such as post-doctoral researchers form the fields of databases, information retrieval, and related fields.
[...]

Web document retrieval

HTML source

```html
<html>
<head>
  <title>Winter School 2013</title>
  <meta name="keywords" content="PROMISE, school, PhD, IR, DB, [...]" />
  <meta name="description" content="PROMISE Winter School 2013, [...]" />
</head>
<body>
  <h1>PROMISE Winter School 2013</h1>
  <h2>Bridging between Information Retrieval and Databases</h2>
  <h3>Bressanone, Italy 4 - 8 February 2013</h3>
  <p>The aim of the PROMISE Winter School 2013 on "Bridging between Information Retrieval and Databases" is to give participants a grounding in the core topics that constitute the multidisciplinary area of information access and retrieval to unstructured, semistructured, and structured information. The school is a week-long event consisting of guest lectures from invited speakers who are recognized experts in the field. The school is intended for PhD students, Masters students or senior researchers such as post-doctoral researchers form the fields of databases, information retrieval, and related fields. [...]
  </p>
</body>
</html>
```
The aim of the PROMISE Winter School 2013 on "Bridging between Information Retrieval and Databases" is to give participants a grounding in the core topics that constitute the multidisciplinary area of information access and retrieval to unstructured, semistructured, and structured information. The school is a week-long event consisting of guest lectures from invited speakers who are recognized experts in the field. The school is intended for PhD students, Masters students or senior researchers such as post-doctoral researchers form the fields of databases, information retrieval, and related fields.
Fielded Language Models
[Ogilvie & Callan, SIGIR’03]

- Build a separate language model for each field
- Take a linear combination of them

\[ P(t|\theta_d) = \sum_{j=1}^{m} \mu_j P(t|\theta_{d,j}) \]

Field language model
Smoothed with a collection model built from all document representations of the same type in the collection

Field weights
\[ \sum_{j=1}^{m} \mu_j = 1 \]
Field Language Model

\[ P(t | \theta_{d_j}) = (1 - \lambda_j)P(t | d_j) + \lambda_j P(t | C_j) \]

- **Empirical field model**: \( \frac{n(t, d_j)}{|d_j|} \)
- **Collection field model**: \( \frac{\sum_d n(t, d_j)}{\sum_d |d_j|} \)
- **Smoothing parameter**: \( \lambda_j \)
- **Maximum likelihood estimates**:\( \frac{\sum_d n(t, d_j)}{\sum_d |d_j|} \)


Fielded Language Models

Parameter estimation

- Smoothing parameter
  - Dirichlet smoothing with avg. representation length

- Field weights
  - Heuristically (e.g., proportional to the length of text content in that field)
  - Empirically (using training queries)
    - Computationally intractable for more than a few fields
Example

\[ q = \{ \text{IR, winter, school} \} \]
fields = \{ title, meta, headings, body \}
\[ \mu = \{ 0.2, 0.1, 0.2, 0.5 \} \]

\[
P(q|\theta_d) = P(\text{“IR”}|\theta_d) \cdot P(\text{“winter”}|\theta_d) \cdot P(\text{“school”}|\theta_d)
\]

\[
P(\text{“IR”}|\theta_d) = 0.2 \cdot P(\text{“IR”}|\theta_{d_{title}})
+ 0.1 \cdot P(\text{“IR”}|\theta_{d_{meta}})
+ 0.2 \cdot P(\text{“IR”}|\theta_{d_{headings}})
+ 0.2 \cdot P(\text{“IR”}|\theta_{d_{body}})
\]
Use case
Entity retrieval in RDF data

Audi A4
From Wikipedia, the free encyclopedia

The Audi A4 is a line of compact executive cars produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group.

The A4 has been built in four generations and is based on Volkswagen's B platform. The first generation A4 succeeded the Audi 80. The automaker's internal numbering treats the A4 as a continuation of the Audi 80 lineage, with the initial A4 designated as the B5-series, followed by the B6, B7, and the current B8. The B8 A4 is built on the Volkswagen Group MLB platform shared with many other Audi models and potentially one Porsche model within Volkswagen Group.[2]

The Audi A4 automobile layout consists of a longitudinally oriented engine at the front, with transaxle-type transmissions mounted at the rear of the engine. The cars are front-wheel drive, or on some models, "quattro" all-wheel drive.

The A4 is available as a saloon/sedan and estate/wagon. The second (B6) and third generations (B7) of the A4 also had a convertible version, but the B8 version of the convertible became a variant of the Audi A5 instead as Audi got back into the compact executive coupé segment. The facebook fans of the Audi A4 page are more than 870,000.
# Use case

## Entity retrieval in RDF data

<table>
<thead>
<tr>
<th>foaf:name</th>
<th>Audi A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:label</td>
<td>Audi A4</td>
</tr>
<tr>
<td>rdfs:comment</td>
<td>The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group. The A4 has been built [...]</td>
</tr>
<tr>
<td>dbpprop:production</td>
<td>1994</td>
</tr>
<tr>
<td></td>
<td>2001</td>
</tr>
<tr>
<td></td>
<td>2005</td>
</tr>
<tr>
<td></td>
<td>2008</td>
</tr>
<tr>
<td>rdf:type</td>
<td>dbpedia-owl:MeanOfTransportation</td>
</tr>
<tr>
<td></td>
<td>dbpedia-owl:Automobile</td>
</tr>
<tr>
<td>dbpedia-owl:manufacturer</td>
<td>dbpedia:Audi</td>
</tr>
<tr>
<td>dbpedia-owl:class</td>
<td>dbpedia:Compact_executive_car</td>
</tr>
<tr>
<td>owl:sameAs</td>
<td>freebase:Audi_A4</td>
</tr>
<tr>
<td>is dbpedia-owl:predecessor of</td>
<td>dbpedia:Audi_A5</td>
</tr>
<tr>
<td>is dbpprop:similar of</td>
<td>dbpedia:Cadillac_BLS</td>
</tr>
</tbody>
</table>
Hierarchical Entity Model
[Neumayer et al., ECIR’12]

- Number of possible fields is huge
  - It is not possible to optimise their weights directly

- Entities are sparse w.r.t. different fields
  - Most entities have only a handful of predicates

- Organise fields into a 2-level hierarchy
  - Field types (4) on the top level
  - Individual fields of that type on the bottom level

- Estimate field weights
  - Using training data for field types
  - Using heuristics for bottom-level types
The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group. The A4 has been built [...]
Hierarchical Entity Model
[Neumayer et al., ECIR’12]

\[ P(t \mid \theta_d) = \sum_F P(t \mid F, d) P(F \mid d) \]

**Term importance**

\[ P(t \mid F, d) = \sum_{d_f \in F} P(t \mid d_f, F) P(d_f \mid F, d) \]

**Field type importance**

Taken to be the same for all entities
\[ P(F \mid d) = P(F') \]

**Term generation**

Importance of a term is jointly determined by
the field it occurs as well as all fields of that

type (smoothed with a coll. level model)

\[ P(t \mid d_f, F) = (1 - \lambda)P(t \mid d_f) + \lambda P(t \mid \theta_{d_F}) \]

**Field generation**
Field generation

- Uniform
  - All fields of the same type are equally important

- Length
  - Proportional to field length (on the entity level)

- Average length
  - Proportional to field length (on the collection level)

- Popularity
  - Number of documents that have the given field
Comparison of models

Unstructured document model

Fielded document model

Hierarchical document model
Use case
Finding movies in IMDB data

The Transporter (2002)
PG-13 92 min - Action | Crime | Thriller - 11 October 2002 (USA)

Your rating: ★★★★★★★★★ -/10
Ratings: 6.7/10 from 133,802 users  Metascore: 51/100
Reviews: 428 user | 154 critic | 27 from Metacritic.com

This film is about a man whose job is to deliver packages without asking any questions. Complications arise when he breaks those rules.

Directors: Louis Leterrier, Corey Yuen
Writers: Luc Besson, Robert Mark Kamen
Stars: Jason Statham, Qi Shu and Matt Schulze | See full cast and crew
Use case
Finding movies in IMDB data

<title>The Transporter</title>
<year>2002</year>
<language>English</language>
<genre>Action</genre>
<genre>Crime</genre>
<genre>Thriller</genre>
<country>USA</country>
<actors>
  <actor>Jason Statham</actor>
  <actor>Matt Schulze</actor>
  <actor>François Berléand</actor>
  <actor>Ric Young</actor>
  <actress>Qi Shu</actress>
</actors>
<team>
  <director>Louis Leterrier</director>
  <director>Corey Yuen</director>
  <writer>Luc Besson</writer>
  <writer>Robert Mark Kamen</writer>
  <producer>Luc Besson</producer>
  <cinematographer>Pierre Morel</cinematographer>
</team>
Probabilistic Retrieval Model for Semistructured data

[Kim et al., ECIR’09]

- Find which document field each query term may be associated with

- Extending [Ogilvie & Callan, SIGIR’03]

\[
P(t|\theta_d) = \sum_{j=1}^{m} \mu_j P(t|\theta_{d_j})
\]

Mapping probability
Estimated for each query term

\[
P(t|\theta_d) = \sum_{j=1}^{m} P(d_j|t)P(t|\theta_{d_j})
\]
PRMS
Mapping probability

\[ P(t|C_j) = \frac{\sum_d n(t, d_j)}{\sum_d |d_j|} \]

**Term likelihood**
Probability of a query term occurring in a given field type

\[ P(d_j|t) = \frac{P(t|d_j)P(d_j)}{P(t)} \]

\[ \sum_{d_k} P(t|d_k)P(d_k) \]

**Prior field probability**
Probability of mapping the query term to this field before observing collection statistics
Query: meg ryan war

d_j  P(t|d_j)
cast  0.407
team  0.382
title  0.187

d_j  P(t|d_j)
cast  0.601
team  0.381
title  0.017

d_j  P(t|d_j)
genre  0.927
title  0.070
location  0.002
Part 2

Query structure
Structured query representations

- Query may have a semistructured representation, i.e., multiple fields

- Examples
  - TREC Genomics track
    - (1) gene name, (2) set of symbols
  - TREC Enterprise track, document search task
    - (1) keyword query, (2) example documents
  - INEX Entity track
    - (1) keyword query, (2) target categories, (3) example entities
  - TREC Entity track
    - (1) keyword query, (2) input entity, (3) target type
Use case
Enterprise document search (TREC 2007)

- Task: create an overview page on a given topic
  - Find documents that discuss the topic in detail

```
<topic>
  <num>CE-012</num>
  <query>cancer risk</query>
  <narr>
    Focus on genome damage and therefore cancer risk in humans.
  </narr>
  <page>CSIRO145-10349105</page>
  <page>CSIRO140-15970492</page>
  <page>CSIRO139-07037024</page>
  <page>CSIRO138-00801380</page>
</topic>
```
DNA Doctor: Catalyst, ABC interview

In this video CSIRO’s Dr Michael Fenech says that damage to the genome is a fundamental disease that can be diagnosed and treated.

(8:00)

Dr Michael Fenech says we should consider damage to the genome as a fundamental disease that can be diagnosed and treated.

In this video ABC Reporter for the television program Catalyst, Mr Paul Willis, acts a guinea pig to test Dr Fenech’s theories.

The video also features an interview with Professor Bruce Armstrong at the University of Sydney, Sydney, NSW, Australia, about the likelihood of being able to repair our genomes.

CSIRO has completed negotiations with a private company to make the genome health analysis test described in this Catalyst interview available to the general public on a commercial basis together with advice on dietary patterns and/or supplements that may assist in prevention of DNA damage.

The launch of the Reach 100 clinic in early July 2007, highlighted the role of preventative health and dietary methods of reducing cancer risk factors.

Learn more about CSIRO’s work Preventing diseases and detecting them sooner.
Query modeling

- Aims
  - Expand the original query with additional terms
  - Assign the probability mass non-uniformly

\[
P(d|q) \propto P(d) \prod_{t \in q} P(t|\theta_d)^{n(t,q)}
\]

\[
\log P(d|q) \propto \log P(d) + \sum_{t \in q} P(t|\theta_q) \log P(t|\theta_d)
\]
Retrieval model

- Maximizing the query log-likelihood provides the same ranking as minimizing KL-divergence
  - Assuming uniform document priors

\[ P(t|\theta_d) \quad KL(\theta_q||\theta_d) \quad P(t|\theta_q) \]

**Document model**

**Query model**
Estimating the query model

- Baseline maximum-likelihood

\[ P(t|\theta_q) = P(t|q) = \frac{n(t,q)}{|q|} \]

- Query expansion using relevance models [Lavrenko & Croft, SIGIR’01]

\[ P(t|\theta_q) = (1 - \lambda)P(t|\hat{q}) + \lambda P(t|q) \]

\[ P(t|\hat{q}) \approx \frac{P(t, q_1, \ldots, q_k)}{\sum_{t'} P(t', q_1, \ldots, q_k)} \]

Expanded query model
Based on term co-occurrence statistics
Sampling from examples
[Balog et al., SIGIR’08]

\[ P(t|S) = \sum_{d \in S} P(t|d)P(d|S) \]

Term importance

\[ P(t|\hat{q}) = \frac{P(t|S)}{\sum_{t' \in K} P(t'|S)} \]

Document importance

\[ d \]

sample documents

\[ P(d|S) \]

\[ d \]

\[ P(t|d) \]

\[ t \]

\[ P(t|S) \]

sampling distribution

top K terms

\[ P(t|\hat{q}) \]

expanded query
Sampling from examples

Importance of a sample document

- **Uniform** \( P(d|S) = 1/|S| \)
  - All sample document are equally important

- **Query-biased** \( P(d|S) \propto P(d|q) \)
  - Proportional to the document’s relevance to the (original) query

- **Inverse query-biased** \( P(d|S) \propto 1 - P(d|q) \)
  - Reward documents that bring in new aspects (not covered by the original query)

\[
P(d|S) = \frac{1}{|S|} \quad P(d|S) \propto P(d|q) \quad P(d|S) \propto 1 - P(d|q)
\]
Sampling from examples

Estimating term importance

- Maximum-likelihood estimate
  \[ P(t|d) = \frac{n(t, d)}{|d|} \]

- Smoothed estimate
  \[ P(t|d) = P(t|\theta_d) = (1 - \lambda)P(t|d) + \lambda P(t|C) \]

- Ranking function by \[\textbf{[Ponte, 2000]}\]
  \[ P(t|d) = s(t)/\sum_{t'} s(t') \quad s(t) = \log \frac{P(t|d)}{P(t|C)} \]
Use case
Entity retrieval in Wikipedia (INEX 2007-09)

- Given a query, return a ranked list of entities
  - Entities are represented by their Wikipedia page

- Entity search
  - Topic definition includes target categories

- List search
  - Topic definition includes example entities
**Titanic (1997 film)**

From Wikipedia, the free encyclopedia

*Titanic* is a 1997 American epic romance and disaster film directed, written, co-produced, and co-edited by James Cameron. A fictionalized account of the sinking of the RMS *Titanic*, it stars Leonardo DiCaprio as Jack Dawson and Kate Winslet as Rose DeWitt Bukater, members of different social classes who fall in love aboard the ship during its ill-fated maiden voyage. Although the central roles and love story are fictitious, some characters are based on genuine historical figures. Gloria Stuart portrays the elderly Rose, who narrates the film in a modern-day framing device, and Billy Zane plays Cal Hockley, the overbearing fiancé of the younger Rose. Cameron saw the love story as a way to engage the audience with the real-life tragedy.

Production on the film began in 1995, when Cameron shot footage of the actual *Titanic* wreck. The modern scenes were shot on board the *Akademik Mstislav Keldysh*, which Cameron had used as a base when filming the actual wreck. A reconstruction of the *Titanic* was built at Playas de Rosarito, Baja California, and scale models and computer-generated imagery were also used to recreate the sinking. The film was partially funded by Paramount Pictures and 20th Century Fox – respectively, its American and international distributor – and at the time, it was the most expensive film ever made, with an estimated budget of US$200 million.[3][4][5][6]

The film was originally scheduled to open on July 2, 1997, however, post-production delays pushed back its release to December 19 instead.[7] *Titanic* was an enormous critical and commercial success. It was nominated for fourteen Academy Awards, eventually winning eleven, including Best Picture and Best Director. It became the highest-grossing film of all time, with a worldwide gross of over $1.8 billion, and remained so for twelve years until Cameron's next directorial effort, *Avatar*, surpassed it in 2010.[9][10] *Titanic* also has been ranked as the sixth best epic film of all time in AFI's 10 Top 10 by the American Film Institute.[11] The film is due for theatrical re-release in 2012 after Cameron completes its conversion into 3-D.[12]
Example query

<title>Movies with eight or more Academy Awards</title>
<categories>
  <category id="45168">best picture oscar</category>
  <category id="14316">british films</category>
  <category id="2534">american films</category>
</categories>
Using categories for retrieval

- As a separate document field
- Filtering
- Similarity between target and entity categories
  - Set similarity metrics
  - Content-based (concatenating category contents)
  - Lexical similarity of category names
Also related to categories

- Category expansion
  - Based on category structure
  - Using lexical similarity of category names
  - Ontology-base expansion

- Generalisation
  - Automatic category assignment
Modeling terms and categories

[Balog et al., TOIS’11]

\[ P(e|q) \propto P(q|e) P(e) \]

\[ P(q|e) = (1 - \lambda) P(\theta_T^q | \theta_T^e) + \lambda P(\theta_C^q | \theta_C^e) \]
Part 3
Contextual structures
Other types of structure

- Link structure
- Linguistic structure
- Social structures
Link structure
Link structure

- **Aim**
  - High number of inlinks from pages relevant to the topic and not many incoming links from other pages

- Retrieve an initial set of documents, then rerank

\[
P(d|q) \propto P(q|d)P(d)
\]

**Document prior**
- Probability of the document being relevant to any query
Document link degree

[Kamps & Koolen, ECIR’08]

Local indegree
Number of incoming links from within the top ranked documents retrieved for $q$

Global indegree
Number of incoming links from the entire collection

$$P(d) \propto 1 + \frac{I_{local}(d)}{1 + I_{global}(d)}$$
Relevance propagation
[Tsikrika et al., INEX’06]

- Model a user (random surfer) that after seeing the initial set of results
  - Selects one document and reads its description
  - Follows links connecting entities and reads the descriptions of related entities
  - Repeats it $N$ times

\[
P_0(d) = P(q|d)
\]

\[
P_i(d) = P(q|d)P_{i-1}(d) + \sum_{d' \rightarrow d} (1 - P(q|d')) P(d'|d)P_{i-1}(d')
\]

The probability of staying at the node equals to its relevance to the query

Outgoing links from $d'$ to $d$

Transition probabilities set to uniform
Relevance propagation
[Tsikrika et al., INEX’06]

- Weighted sum of probabilities at different steps

\[ P(d) \propto \mu_0 P_0(d) + (1 - \mu_0) \sum_{i=1}^{N} \mu_i P_i(d) \]
Translation Model

[Berger & Lafferty, SIGIR’99]

\[ P(q|d) = \prod_{t \in q} \left( \sum_{w \in V} P(t|w) P(w|\theta_d) \right)^{n(t,q)} \]

- Obtaining the translation model
  - Exploiting WordNet word co-occurrences [Cao et al., SIGIR’05]

Translation model
Probability that word \( w \) can “semantically translated” to word \( q_i \)
Use case

Expert finding

- Given a keyword query, return a ranked list of people who are experts on the given topic
- Content-based methods can return the most knowledgeable persons
- However, when it comes to contacting an expert, social and physical proximity matters
Expert profile

<person>
    <anr>710326</anr>
    <name>Toine M. Bogers</name>
    <name>Toine Bogers</name>
    <name>A. M. Bogers</name>
    <job>PhD student</job>
    <faculty>Faculty of Humanities</faculty>
    <department>Department of Communication and Information Sciences</department>
    <room>Room D 348</room>
    <address>P.O. Box 90153, NL-5000 LE Tilburg, The Netherlands</address>
    <tel>+31 13 466 245</tel>
    <fax>+31 13 466 289</fax>
    <homepage>http://ilk.uvt.nl/~toine</homepage>
    <email>A.M.Bogers@uvt.nl</email>

    <publications>
        <publication arno-id="1212"/>
            <title>Design and Implementation of a University-wide Expert Search Engine</title>
            <author>R. Liebregts and T. Bogers</author>
            <year>2009</year>
            <booktitle>Proceedings of the 31st European Conference on Information Retrieval</booktitle>
        </publication>
        [...]
    </publications>
</person>
User-oriented model for EF
[Smirnova & Balog, ECIR’11]

\[ S(e|u, q) = (1 - \lambda)K(e|u, q) + \lambda T(e|u) \]

- **Knowledge gain**: Difference between the knowledge of the expert and that of the user on the query topic
- **Contact time**: Distance between the user and the expert in a social graph
Social structures

Organisational hierarchy

<person>
  <anr>710326</anr>
  <name>Toine M. Bogers</name>
  [...] 
  <faculty>Faculty of Humanities</faculty>
  <department>Department of Communication and Information Sciences</department>
  [...] 
</person>
Social structures

Geographical information

<person>
    <anr>710326</anr>
    <name>Toine M. Bogers</name>
    [...]  
    <room>Room D 348</room>
    [...] 
</person>
Social structures

Co-authorship

<person>
  <anr>710326</anr>
  <name>Toine M. Bogers</name>
  [...]
 <publications>
  <publication arno-id="1212">
   <title>
    Design and Implementation of a University-wide Expert Search Engine
   </title>
   <author>
    R. Liebregts and T. Bogers
   </author>
   <year>2009</year>
   <booktitle>
    Proceedings of the 31st European Conference on Information Retrieval
   </booktitle>
  </publication>
  [...]
 </publications>
</person>
Social structures

Combinations

- University
- Department
- Faculty
- Building
- Campus
- Floor
- Toine M. Bogers
  - Faculty
  - Department
  - Building
  - Campus
Summary

- Different types of structure
  - Document structure
  - Query structure
  - Contextual structures

- Probabilistic IR and statistical Language Models yield a principled framework for representing and exploiting these structures
Questions?