Dealing with the Deep Web and all its Quirks

(joint work with M. Bienvenu, D. Deutch, D. Martinenghi, and F. Suchanek)

PIERRE SENELLART
Definition (Deep Web, Hidden Web, Invisible Web)

All the content on the Web that is not directly accessible through hyperlinks. In particular: HTML forms, Web services.

Size estimate: 500 times more content than on the surface Web! [BrightPlanet, 2001]. Hundreds of thousands of deep Web databases [Chang et al., 2004]
Sources of the Deep Web

Example

- Yellow Pages and other directories;
- Library catalogs;
- Weather services;
- Real-estate agencies;
- etc.

... but also lots of information available on the surface Web, but that may be interesting to retrieve from the deep Web:

- more structured
- easier to retrieve the information of interest
- less network accesses to crawl the whole database
Numerous works on **form understanding and information extraction** from the deep Web [He et al., 2007, Varde et al., 2009, Khare et al., 2010]

Formal models for answering queries under **access pattern restrictions** [Li and Chang, 2001, Calì and Martinenghi, 2008, Calì and Martinenghi, 2010, Benedikt et al., 2012a]

**Siphoning** of hidden Web databases [Barbosa and Freire, 2004, Jin et al., 2011, Sheng et al., 2012]

Those works ignore lots of **quirky dimensions** of deep Web interfaces

Here: towards a more comprehensive framework for **deep Web modeling and querying**
Outline

Introduction

Deep Web Quirks

Towards a Data Model and Query Language

Problems of Interest

Conclusions
Views

Deep Web sources offer views over (most often relational) data, through, at the very least:

- **selection** (depending on user’s query, or implicit in the service), in particular inequalities
- **projection** (not available attributes are exported by a given service)

And also (but less critically):

- **joins** (quite common in a Web application – but from an outsider’s perspective, often enough to see the result of a join as the relation of interest)
- union, intersection, difference, etc. (relatively rare)
- **aggregation** (usually not the most important part of the service)
- more **complex** processing (rare in practice)
Limited access patterns

Australian Yellow Pages search form:
Limited access patterns

Australian Yellow Pages search form:

Required attributes, dependencies between attributes of the form, etc.
Ranking of results

IMDb advanced search sort criteria:

Different possible sort criteria, some according to non-exported attributes
Paging in IMDb:

Each page of results requires a separate network access, and therefore has a cost.
Overflow

What you get when you try to access the 100,001-th result to an IMDb advanced query:

Error

Sorry, IMDb does not serve more than 100000 results for any query. (You asked for results starting from 100001)

Only a (top-ranked) subset of the results is available for each access
Policy limitations

Twitter API rate limitation:

REST API Rate Limiting

The default rate limit for calls to the REST API varies depending on the authorization method being used and whether the method itself requires authentication.

- Unauthenticated calls are permitted 150 requests per hour. Unauthenticated calls are measured against the public facing IP of the server or device making the request.
- OAuth calls are permitted 350 requests per hour and are measured against the oauth_token used in the request.

Limited rate of queries per minute, hour, query... Several services of the same source may share the same limits.
Incomplete information: Projection

Several views of the same information on IMDB:
Incomplete information: Projection

Several views of the same information on IMDB:

1. *It's a Wonderful Life* (1946)
   - aka "Frank Capra's It's a Wonderful Life" - USA (*complete title*)
   - aka "La vie est belle" - Belgium (*French title*), Canada (*French title*), France
   - aka "IQué bello es vivir!" - Peru (*imdb display title*), Spain
   - aka "Ist das Leben nicht schön?" - Austria (*TV title*), West Germany (*TV title*)
   - aka "Qué bello es vivir!" - Uruguay
   - aka "A Felicidade Não Se Compra" - Brazil
   - aka "Az élet csodaszép" - Hungary
   - aka "Det er herligt at leve" - Denmark
   - aka "Divan život" - Serbia
   - aka "Divan zivot" - Yugoslavia (*Croatian title*) (*imdb display title*)
   - aka "Do Céu Caiu Uma Estrela" - Portugal
   - aka "Inmeillinen on elämä" - Finland
   - aka "La vita è meravigliosa" - Italy
   - aka "Livet är underbart" - Sweden
   - aka "Livet er vidunderlig" - Norway (*imdb display title*)
   - aka "Mens, durf te leven" - Netherlands (*informal literal title*)
   - aka "Mia yperohi zoi" - Greece (*transliterated ISO-LATIN-1 title*)
   - aka "O viata minunata" - Romania (*imdb display title*)
   - aka "Qué bello es vivir" - Argentina
   - aka "Que bonic és viure!" - Spain (*Catalan title*)
   - aka "Que la vie est belle" - Belgium (*French title*)
   - aka "Sahane hayat" - Turkey (*Turkish title*) (*DVD title*)
   - aka "Subarashiki kana, jinsei!" - Japan
   - aka "To wspaniale życie" - Poland
   - aka "Wat een mooi leven" - Belgium (*Flemish title*)
   - aka "Zycie jest cudowne" - Poland
Incomplete information: Projection

Several views of the same information on IMDB:

1. *It's a Wonderful Life* (1946)
   - Rating: 8.7/10
   - Description: An angel helps a compassionate but despairingly frustrated businessman by showing what life would have been like if he never existed.
   - Director: Frank Capra
   - Cast: James Stewart, Donna Reed, Lionel Barrymore
   - Genres: Drama, Fantasy
   - Duration: 130 mins.

2. *It Happened One Night* (1934)
   - Rating: 8.3/10
   - Description: A spoiled heiress, running away from her family, is helped by a man who's actually a reporter looking for a story.
   - Director: Frank Capra
   - Cast: Clark Gable, Claudette Colbert, Walter Connolly
   - Genres: Comedy, Romance
   - Duration: 105 mins.

3. *Mr. Smith Goes to Washington* (1939)
   - Rating: 8.4/10
   - Description: A naive man is appointed to fill a vacancy in the US Senate. His plans promptly collide with political corruption, but he doesn't back down.
   - Director: Frank Capra
   - Cast: James Stewart, Jean Arthur, Claude Rains
   - Genres: Comedy, Drama
   - Duration: 129 mins.

Same relation(s), different attributes projected out
Incomplete information: Granularity

Release date API on IMDb:

<table>
<thead>
<tr>
<th>Country</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>20 December 1946</td>
</tr>
</tbody>
</table>

(New York City, New York)

The *granularity* of the presented information may not be the most precise one.
Recency

Savills property search:

Publication time is a special attribute of interest:

- may or may not be exported
- may or may not be queriable (sometimes in a very weird way!)
- often used as a ranking criterion
- granularity plays an important role
- publication date $<$ query date
Uncertainty in the ranking

Amazon Books sorting options:

- Proprietary ranking functions
- Weighted combination of attributes with unknown weights [Soliman et al., 2011]
- Ranking according to an unexported attribute
Dependencies across services

Some of IMDb advanced search options:

**Advanced Title Search**
Want to get a list of comedies from the 1970s that have at least 1000 votes and an average rating of 7.5 or higher? Use Advanced Title Search.

**Advanced Name Search**
Want a list of males in the database who are Virgos and over 6 feet tall? Use Advanced Name Search.

**Collaborations and Overlaps**
Want a list of titles in which both Brad Pitt and George Clooney appeared? Or a list of people who worked on both Forrest Gump and Apollo 13? Try searching Collaborations and Overlaps.

- services of the same source provide different correlated views of the same data
- dependencies (inclusion) across services are common too
- a given service often satisfies some key dependencies
But also...

- **non-conjunctive forms** (common in digital library applications)
- **unknown characteristics** of information retrieval systems (keyword querying vs exact querying, indexing of stop words, stemming used, etc.)
- **intricate interactions** (AJAX autocompletion, submitting a form as a first step before submitting another form, etc.)
- **potential side effects** of a service
Outline

Introduction

Deep Web Quirks

Towards a Data Model and Query Language
  Desiderata
  Example Syntax

Problems of Interest

Conclusions
Outline

Introduction

Deep Web Quirks

Towards a Data Model and Query Language

Desiderata

Example Syntax

Problems of Interest

Conclusions
Features of the query language

What does a user need out of a deep Web query language?

- Selection, projection, joins, union (of different sources)
- Custom ranking
- Top-$k$ results of a query

But also:

- Proper uncertainty management
- Deduplication of query results
- Diversification of query results
- Explanation of query results
Desirable model properties

Declarative framework (specifying what a user wants, not how to retrieve it)

Composability: Web services, queries, materialized views expressible in a common language

Incremental maintenance support

Familiarity with the query language (e.g., relying on SQL when possible)

Cost model for accessing a deep Web source, paging, utilizing a materialized view, etc.
Outline

Introduction

Deep Web Quirks

Towards a Data Model and Query Language
  Desiderata
  Example Syntax

Problems of Interest

Conclusions
Example service: Hotel availability

CREATE VIEW HotelsService1($c,$o) AS
SELECT name, city, price, AvailableRooms, rating, DAY(LastUpdate)
FROM Hotels1
WHERE city=$c
ORDER BY rating DESC
LIMIT $o,10 UP TO 1000

- **Parametrized view** over a (hidden) source relation
- **Main idea**: Reproduce a (possible) SQL implementation of the view
- **Showcased**: selection, projection, access patterns, granularity, ranking, paging, overflow
Example service: Mapping

CREATE VIEW MapService($locX,$locY,$radius, $o) AS
SELECT name, HotelLocX,HotelLocY,
square(HotelLocX-$locX) + square(HotelLocY-$locY) As D
FROM GeoDB
WHERE D < square($radius)
ORDER BY SqrDist ASC
LIMIT $o,10
SELECT Hotels1.name, Hotels2.name
FROM (HotelsService1+HotelsService2+MapService) As H1, (HotelsService1+HotelsService2+MapService) As H2
WHERE H1.city='Istanbul' AND H2.city='Istanbul'
AND H1.rating > 4
AND H2.rating > 4

The “+” operator combines services using any combination of accesses (in particular, union, natural join)
Outline

Introduction

Deep Web Quirks

Towards a Data Model and Query Language

Problems of Interest

Conclusions
Algorithms for, and complexity of, the following problems:

- Given a collection of services, is a query **realizable**? Combines problems from answering queries using views [Halevy, 2001], limited access patterns [Cali and Martinenghi, 2010], feasibility of a ranking function, taking into account overflow...

- What is the **optimal plan** for realizing a query?
  
  **Static plans**: requires a proper query plan (recursive) formalism, and a static cost model
  
  **Dynamic plans**: partial execution and reevaluation of the cost – what is the best access I can do at a given time [Benedikt et al., 2011]
Outline

Introduction

Deep Web Quirks

Towards a Data Model and Query Language

Problems of Interest

Conclusions
Inference of the model from real services

How to automatically infer such a model from real-world forms?

- **Heuristics** to detect paging, overflow, etc.
- Combine classical form understanding and information extraction systems to understand the properties of a service: making assumptions, and then probing to confirm these assumptions [Oita et al., 2012]
- **Software testing** methods to test a wide range of possible combinations of attributes and infer the corresponding behavior of the interface
- Perform **static analysis on client-side code** to detect all such characteristics enforced on the client side [Benedikt et al., 2012b]
- Make use of the different services of the same source to holistically learn their characteristics
Summary and perspectives

- Many **quirky** aspects often ignored but crucial in deep Web services
- A proper query answering system requires consider them together, **not in isolation**
- Towards a **composable, declarative**, model for deep Web querying together with a **cost model**
Summary and perspectives

- Many **quirky** aspects often ignored but crucial in deep Web services
- A proper query answering system requires consider them together, not in isolation
- Towards a **composable, declarative**, model for deep Web querying together with a **cost model**

- Full design of the data and query model
- Characterization of the complexity of the considered problems
- Query planning algorithms


