

Exploring the knowledge in Semi Structured Data Sets with Rich Queries

Jürgen Umbrich
Sebastian Blohm



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft



Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Institut AIFB, Universität Karlsruhe (TH)

Forschungsuniversität • gegründet 1825

Overview

- How to use an annotation engine to extract implicit knowledge encoded in semi structured data sets.
- How to discover, in an automatic way, relation patterns between concepts/categories.
- A framework with support for free text search combined with annotation search
- A user-interface, that hides the complexity of a structured query syntax from the end-user

Motivation: (Semi-) Structured KBs

- Effort to manage unstructured information in (semi-) structured knowledge bases
 - Encyclopedias, like Wikipedia
 - ODP
- Information management is often maintained either manually and/or in a supervised manner.
- KB's reflect the “wisdom of the crowd” and cover a lot of different domains

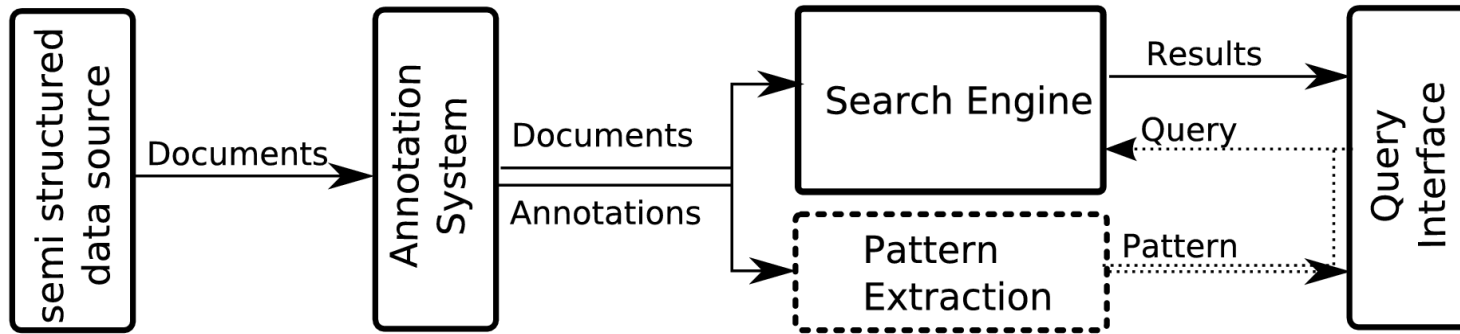
Challenges

- Current approaches to **access** information in semi structured KB
 - Keyword search interface
 - Exploring the data set by article and category links or facets
-> Works well only for small and/or specific data sets
- Inability to incooperate **background knowledge** of users
 - Background knowledge about the domain
 - Knowledge about relations in the result pages

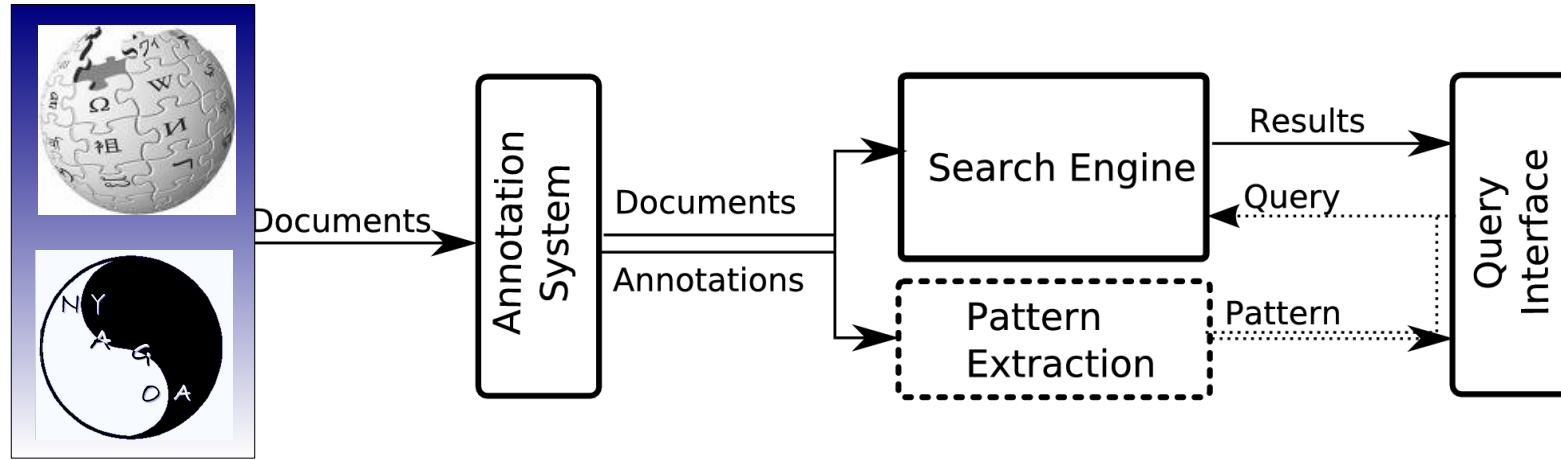
Solution Approach:

- Accessing the KB via freetext search combined with annotation search
 - By extracting implicit knowledge encoded in the KB like **categories and links** between articles,
 - By semantical grounding of extracted information with an ontology
- Support the background knowledge of end-users
 - By automatically extracted relation patterns from the knowledge base

Solution Architecture: Overview

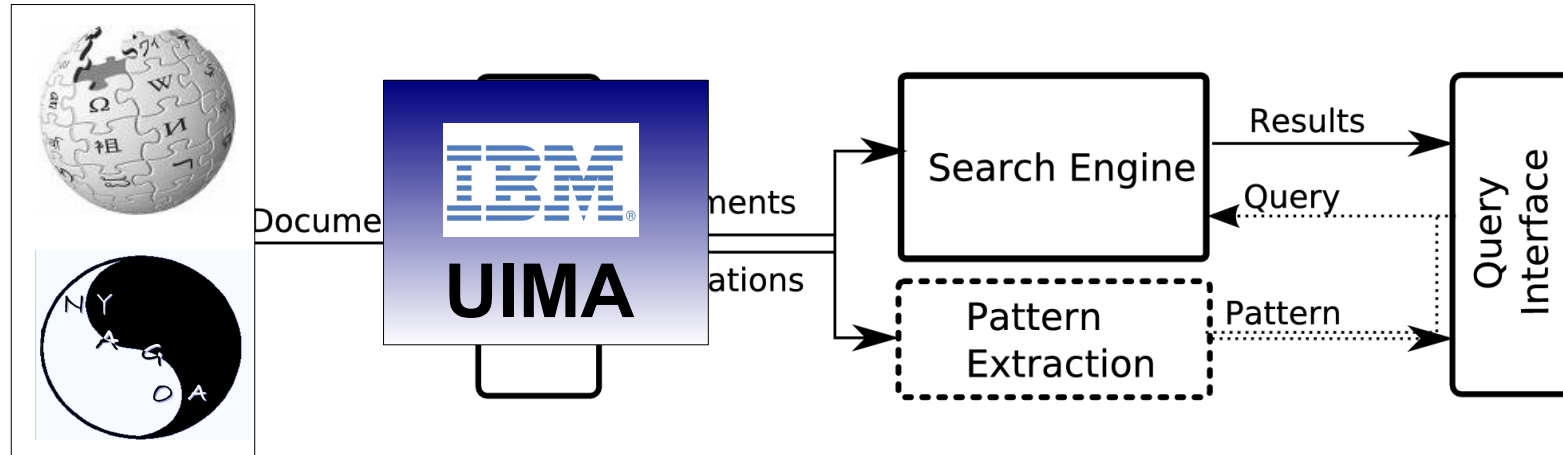


(Semi-)Structured Data Source



- A local **Wikipedia** dump (current from December 2006)
- Wikipedia categories semantically grounded with the **Yago ontology**
 - Mapping between Wikipedia categories and the Yago hierarchy

Annotation System (AS)



- Unstructured Information Management Architecture (UIMA)
- Open source Java framework (Sun Java version)
- Various text analysis engines (TAE) build from scratch
 - Sentence and paragraph splitters, tokenizer ,etc ...
- Easy development of own TAE (in Java)
 - e.g. a simple data annotator or Wikipedia to Yago category mapping

(AS) Article Annotations



article discussion edit this page history

Karl Steinbuch
From Wikipedia, the free encyclopedia

Karl Steinbuch **June 15, 1917** in **Stuttgart-Bad Cannstatt** · **June 4, 2005** in **Ettlingen**) was a German **computer scientist**, **cyberneticist**, and **electrical engineer**. He is one of the pioneers of the German **computer science**, as well as with his **Lernmatrix** an early pioneer of **artificial neural networks**. Steinbuch

Categories: [1917 births](#) | [2005 deaths](#) | [Cyberneticists](#) | [German computer scientists](#) | [German electrical engineers](#) | [German inventors](#) | [Machine learning researchers](#)

Annotation of

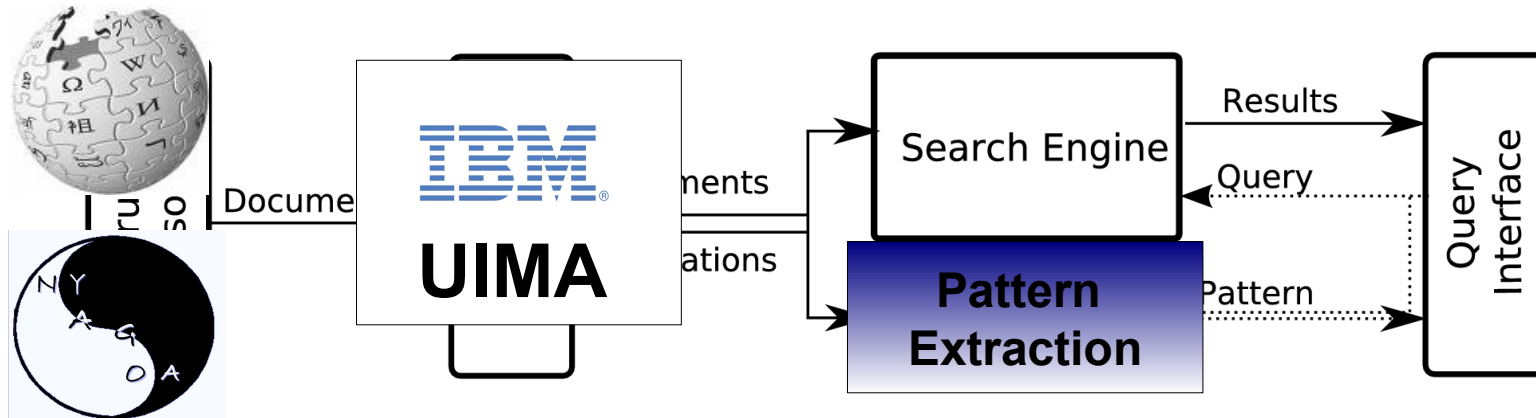
- Title and title occurrence in the article**
- Links and the categories of the linked articles**
- Article categories**

Wikipedia article of "Karl Steinbuch"

■ Semantically enriched annotations

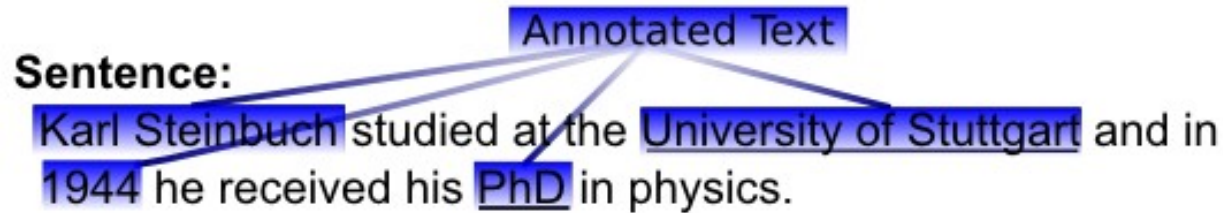
- Utilising the Yago ontology
- Each wikipedia category is modelled as a hierarchy of yago categories e.g. German_computer_scientist ->... -> scientist ->... -> person

Annotation System (AS)



Relation Pattern Extraction

Semi - semantic relation patterns



Extracted Relations: (wikipedia categories grounded to Yago)

- <person> "studied at the" <university>
- <person> "studied at the" "and in" <year>
- <person> "studied at the" "and in" "he received his" <degree>

- Extracting relation between wikipedia categories and their respective Yago categories
 - In sentences and paragraphs
- Relation patterns support query creation in the UI

Crawling - Indexing - Searching

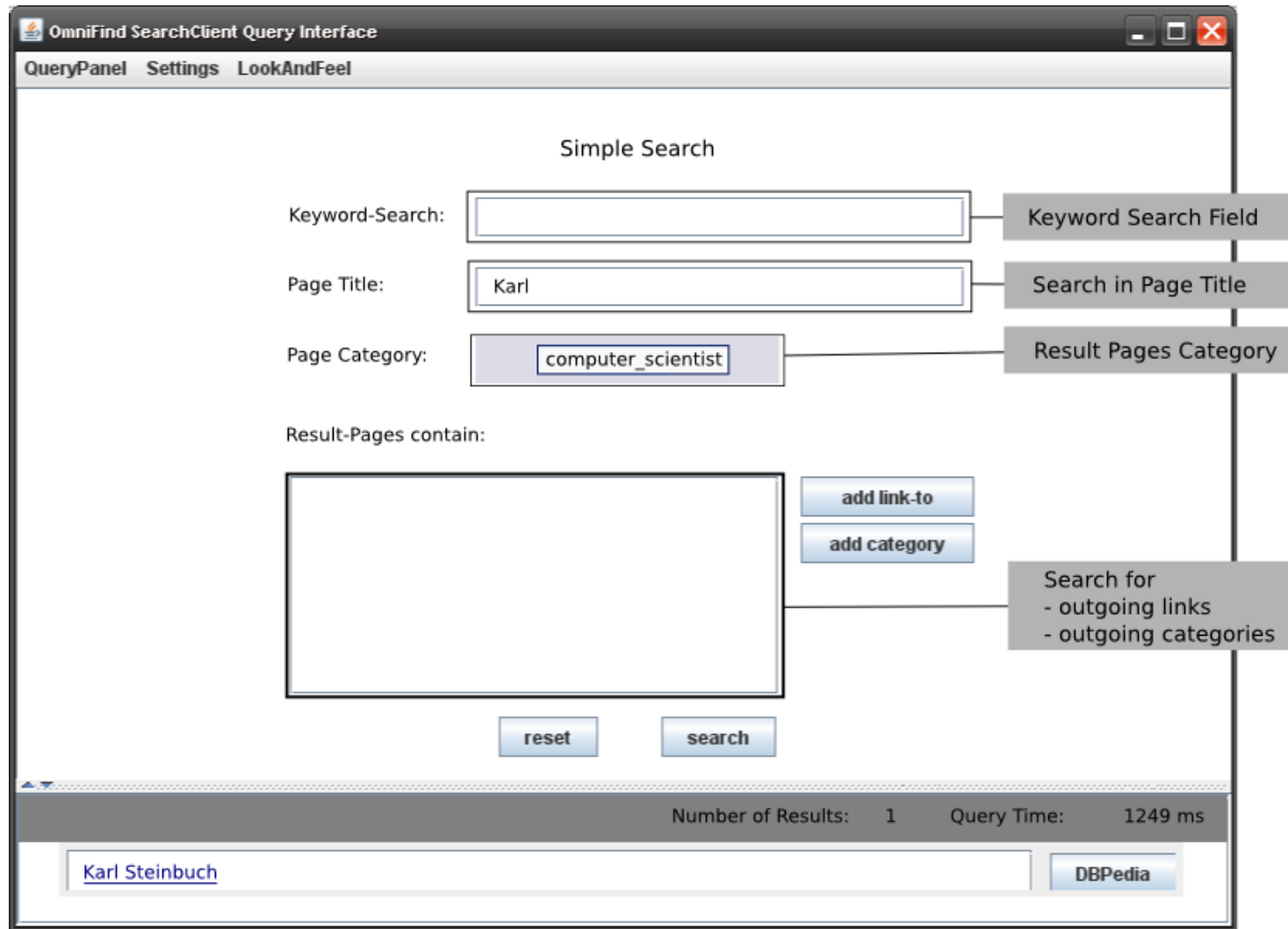


- Integrates UIMA (Version 1.4 , current published Version is 2.x)
- Query functionality to combine text with annotation search
- XML-Fragment or Xpath Syntax



- Hides the complexity of the underlying query syntax
- Two query creation modes (Simple and Advanced)
- Uses known query concepts (e.g. Textfield and DropDown menu)
- Implementation in Java Swing

User Interface: Simple Mode



Omnifind SearchClient Query Interface

QueryPanel Settings LookAndFeel

Simple Search

Keyword-Search:

Page Title:

Page Category:

Result-Pages contain:

Number of Results: 1 Query Time: 1249 ms

[Karl Steinbuch](#)

Keyword Search Field

Search in Page Title

Result Pages Category

Search for
- outgoing links
- outgoing categories

User Interface: Advanced Mode

QueryPanel

Advanced Query Interface

Page OR

Sentence *

Sentence * * * *

Query in XMLF2

```
@xmlf2::<page category="scientist" /> OR <page category="person" />
+ <s> </ title> * <link category="country" >Germany</link> </s>
+ <s> </ title> * studied at" * <link category="university" >Stuttgart</ link>
  * "received" * <link category="degree" /> </s>
```

Query in words:

All pages containing

- pagetype <scientist> OR pagetype <person>
- a sentence with: a <title> WILDCARD a(n) <country> with label "Germany"
- a sentence with: a <title> WILDCARD keyword "studied at" WILDCARD a(n) <university> with label "Stuttgart"
WILDCARD keyword "received" WILDCARD a(n) <degree>

Conclusion

- Automatical extraction of (semi) semantic relation patterns from semi structured KB
 - Using text information extraction tools
- Explore and query semi structured KB's with a combination of freetext and annotation search
 - Document or entity centric search
 - Instance and/or concept search

- Evaluation of the architecture
 - Annotation time (current measures: ~3sec/doc)
 - Indexing time/size
 - Query time related to query complexity
- Automatical discovery of semantic relation patterns (SRP)
 - e.g. *[concept:person] [relation:studiedAt] [concept:university]*
 - Automatically build a concept-relation ontology based on the Wikipedia corpus
- Extending UI interoperability capabilities
 - Support the UI with semantic relation patterns

Exploring the knowledge in Semi Structured Data Sets with Rich Queries

Questions?

■ Contact

juergen@umbrich.net

blohm@aifb.uni-karlsruhe.de

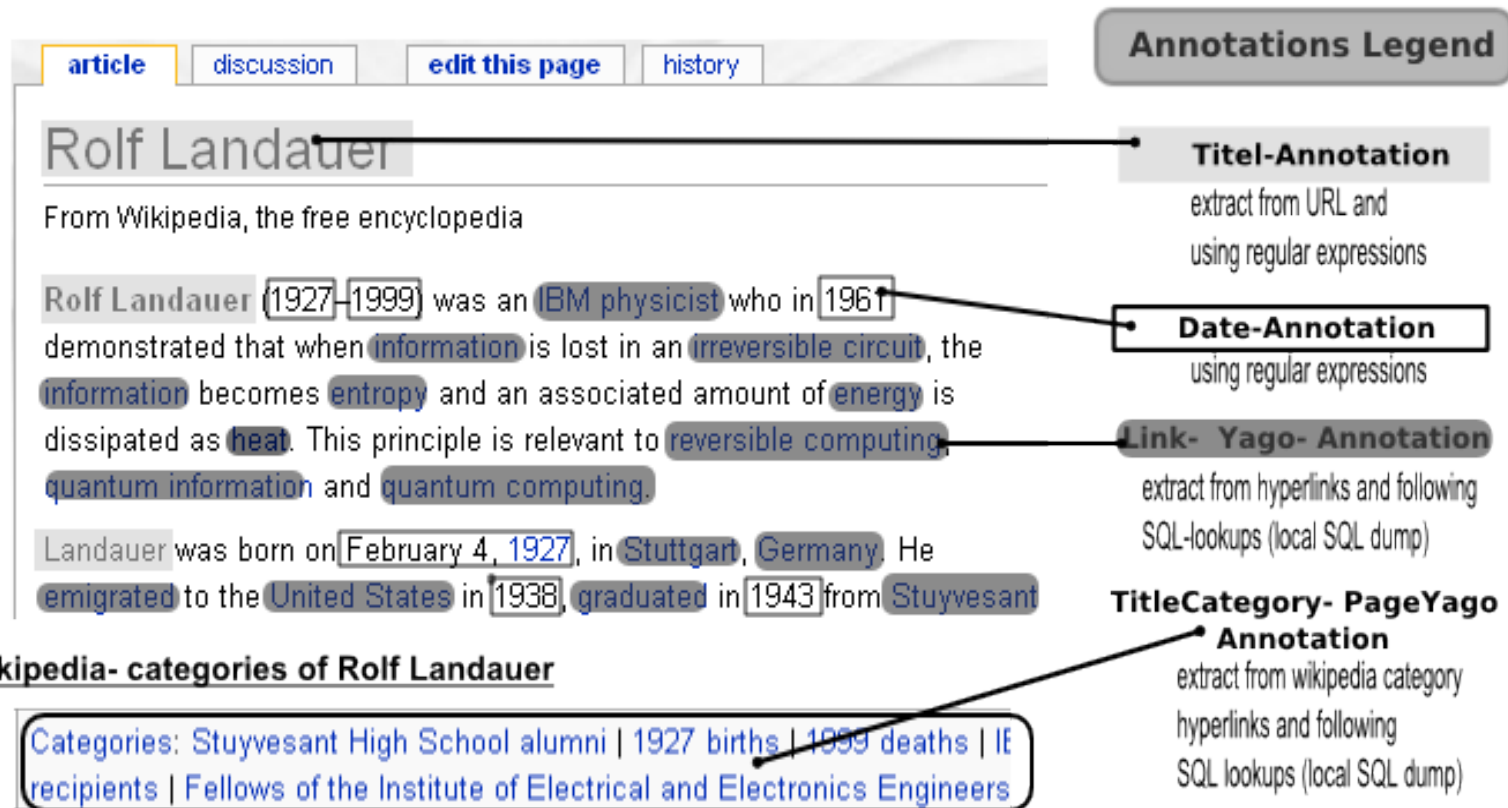
Work was supported by



Article Annotations

- Annotate Wikipedia articles by
 - Title and title occurrences in the article
 - Article categories
 - Links, link title and link categories
 - Date information from various date formats
- Annotation semantically grounded with the Yago ontology
 - Each wikipedia category is modelled as a hierarchy of yago categories
e.g. `american_tennis_player` -> `player` -> `person` -> `causal_agent`

Sample: Annotated Wikipedia article



The image shows a screenshot of a Wikipedia article for Rolf Landauer. The article text is annotated with various colored boxes. A legend on the right side explains the annotations:

- Titel-Annotation**: extract from URL and using regular expressions
- Date-Annotation**: using regular expressions
- Link- Yago- Annotation**: extract from hyperlinks and following SQL-lookups (local SQL dump)
- TitleCategory- PageYago Annotation**: extract from wikipedia category hyperlinks and following SQL lookups (local SQL dump)

The article text includes: "Rolf Landauer (1927-1999) was an IBM physicist who in 1961 demonstrated that when information is lost in an irreversible circuit, the information becomes entropy and an associated amount of energy is dissipated as heat. This principle is relevant to reversible computing, quantum information and quantum computing. Landauer was born on February 4, 1927, in Stuttgart, Germany. He emigrated to the United States in 1938, graduated in 1943 from Stuyvesant High School." Below the article, the categories are listed: "Categories: Stuyvesant High School alumni | 1927 births | 1999 deaths | IF recipients | Fellows of the Institute of Electrical and Electronics Engineers".

OmniFind: Query Syntax

- XML-Fragment or Xpath Syntax
- Access to the KB: API or User-Interface
- Sample Query

@xmlf2::'

```
<page category="scientist" /> OR <page category="person" />
```

```
+ <sentence>
```

```
  </title> * "born in" * <link category="country">Germany</link>
```

```
  </sentence>
```

```
+ <sentence>
```

```
  </title> * "studied at" * <link category="University"/>
```

```
  * <link category="city">Karlsruhe</link>
```

```
  * "received" * <link category="Degree"/>
```

```
</sentence>
```

,

Table of Content

- **Motivation**
- **Architecture**
- **Annotation Engine**
- **User Interface**
- **Future Work**