Entity-Oriented Search

PRI 23/24 · Information Processing and Retrieval
M.EIC · Master in Informatics Engineering and Computation

Sérgio Nunes
Dept. Informatics Engineering
FEUP · U.Porto

Based on Entity-Oriented Search, Springer, K. Balog (2018)

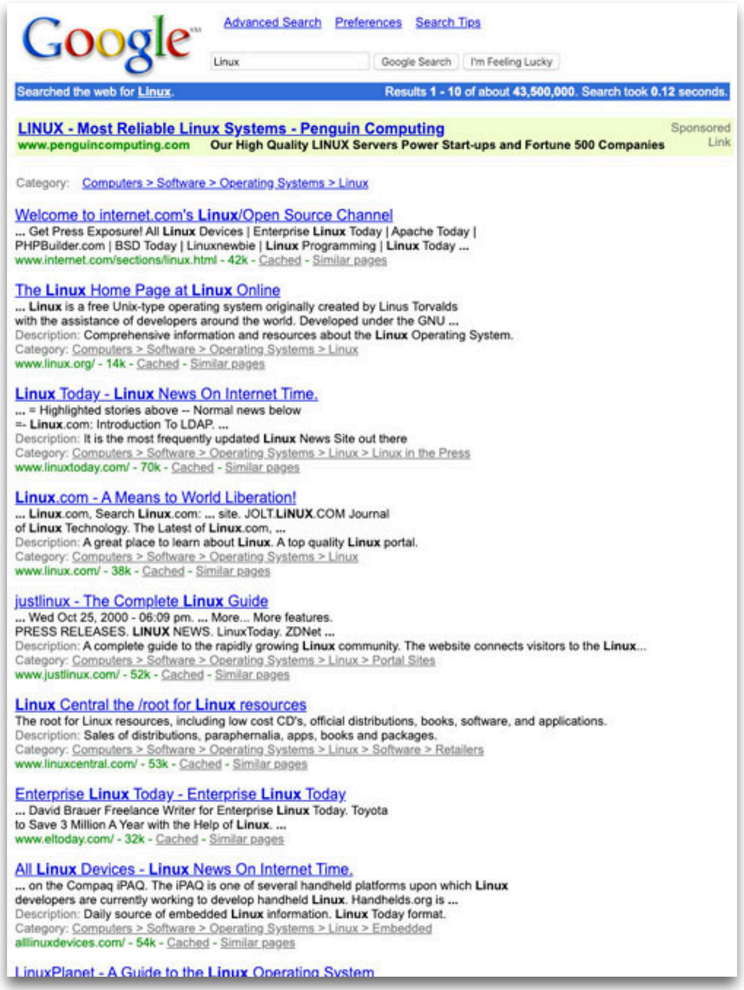
Outline

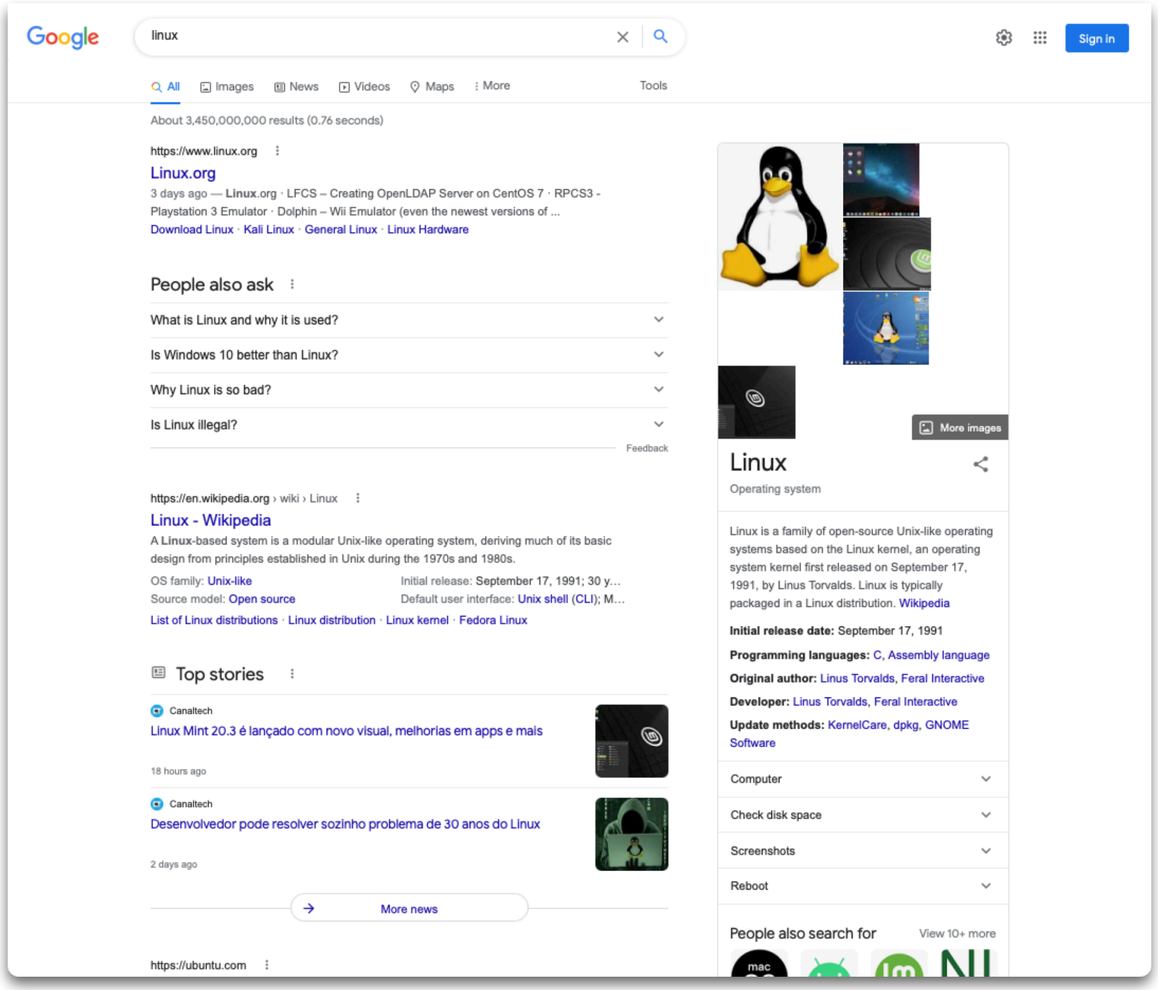
- ➔ Overview of entity-oriented search
- ➔ Types of data sources
- ➔ Entity ranking
- ➔ Entity linking (briefly)

Entity-Oriented Search

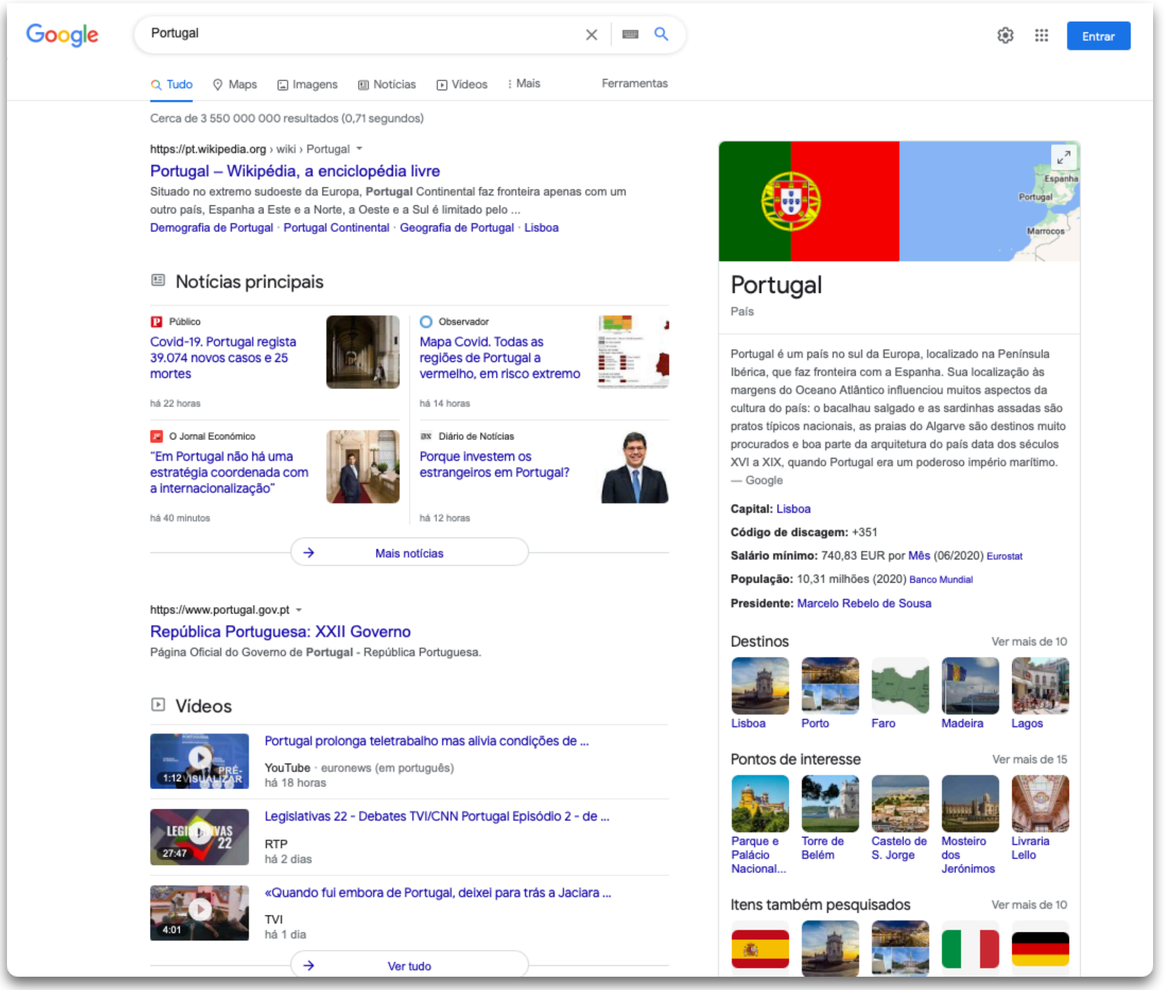
Context

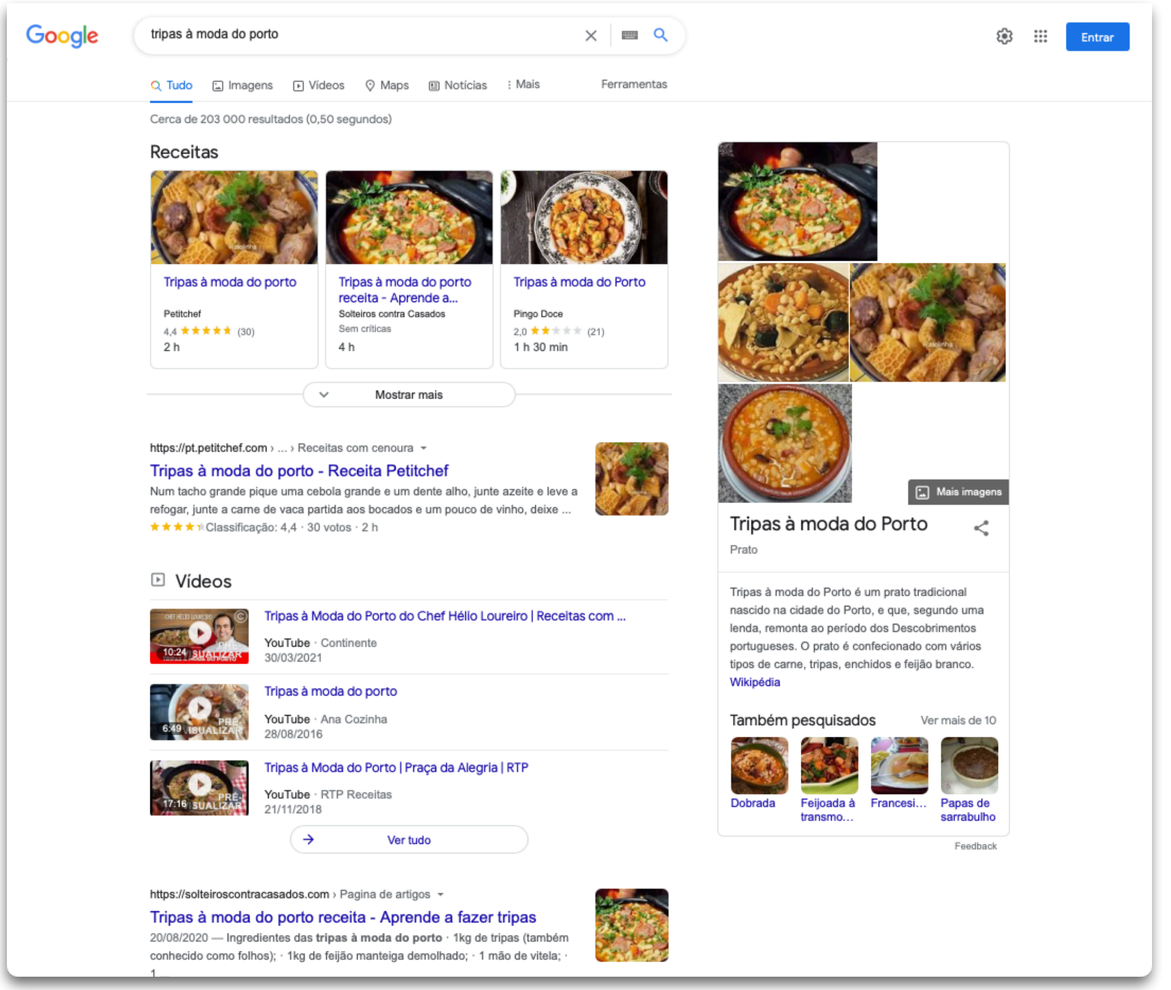
- ➔ Search is pervasive — on the web, on websites, on applications, on devices.
- ➔ Users expect to find a search box to help them find the information they need.
- ➔ Information retrieval deals with matching information needs with information objects.
- ➔ Queries are the expression of information needs.
- ➔ Queries may range from simple keywords [ps5 games] to full sentences ["Where can I buy a Playstation 5?"].
- ➔ The search engine responds with a ranked list of information objects.
- ➔ Traditionally these information objects were documents, but today's search engines return rich results pages beyond the classic "ten blue links" — *semantic search*.



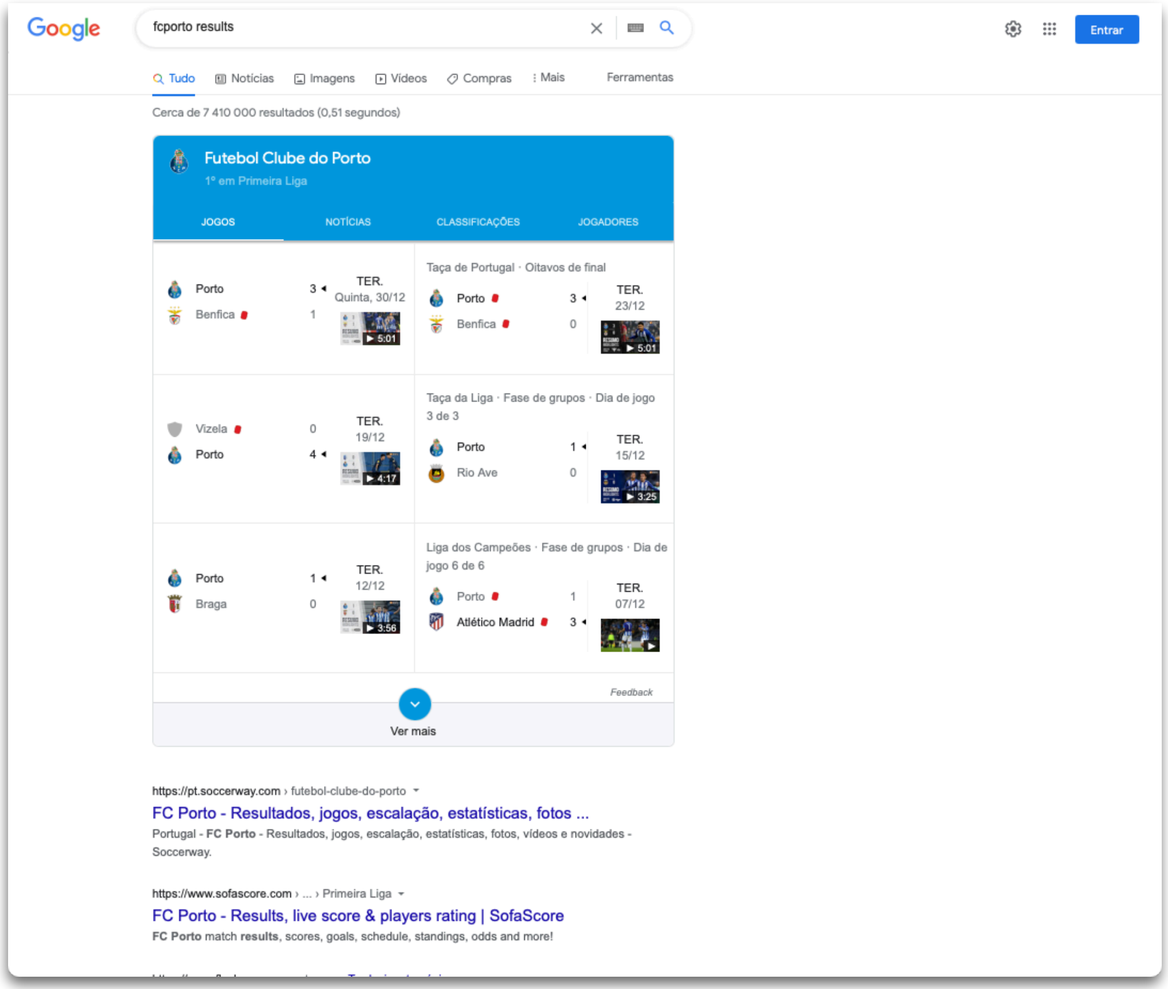


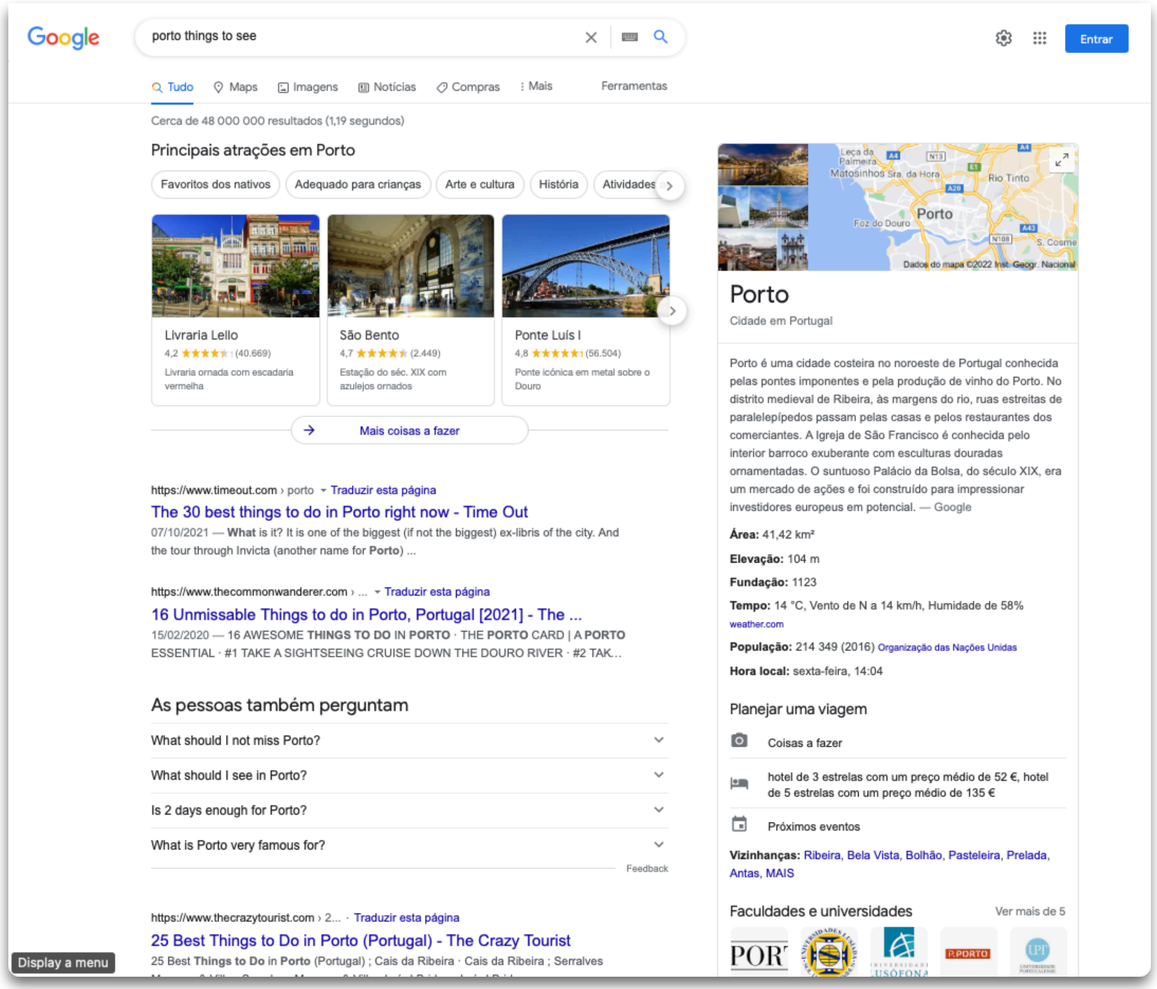
Google search for ["linux"] circa 2000 and today (2021).





Rich search engine results pages (SERPs), containing links to documents, information boxes, direct display of entities, facts, and other structured results.





More examples of "semantic search".

Knowledge Bases and Entities

- ➔ A primary component enabling these advanced search services is the availability of large-scale structured knowledge repositories, called knowledge bases.
- ➔ Knowledge bases organize information around specify things or objects, called "entities".
- ➔ An entity is a uniquely identifiable object or thing, characterized by its name(s), type(s), attributes, and relationships to other entities.
- ➔ Common types of entities include: people, organizations, products, locations, and events.

Properties of Entities

- **Unique identifier**: entities need to be uniquely identifiable. There must be a one-to-one correspondence between each entity identifier and the object it represents. Examples include: username, email address, URI, Wikipedia page ID, etc.
- **Name(s)**: entities are known and referred to by their name. Unlike IDs, names do not uniquely identify entities; multiple entities may share the same name; and the same entity may be known by more than a single name.
- **Type(s)**: entities may be categorized into multiple entities types. Types work as semantic categories that group together entities with similar properties. The set of possible entity type is often organized in a hierarchical structure, e.g. "scientist" is a subtype of "person".
- **Attributes**: the characteristics or features of an entity are described by a set of attributes. Different entities have different sets of attributes. Some of the characteristics may be entities themselves, in this we cases they are not treated as attributes but as relationships, e.g. "place of birth" links to a location entity. Attributes always have literal values and optionally may also include data type information.
- **Relationships**: describe how two entities are associated to each other. Relationships may also be seen as "typed links" between entities.

Representing Properties of Entities

- Information about entities can be represented and stored in semi-structured or in structured form.
  - Wikipedia is an example of a knowledge repository that organizes information about entities and their attributes and relationships in a semi-structured form.
  - To adopt a structured form, a knowledge representation model is necessary. The Resource Description Framework (RDF) is a standard way to describe entities, where an entity is represented as a set of RDF statements.
- A knowledge base (KB) is a structured knowledge repository that contains a set of facts (assertions) about entities.
- Entities in a knowledge base may be seen as nodes in a graph, with relationships between them as edges. Thus, knowledge bases are also referred to as knowledge graphs.

DBpedia Browse using Formats Faceted Browser Sparql Endpoint

About: University of Porto

An Entity of Type: Public university, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

The University of Porto (Universidade do Porto) is a Portuguese public university located in Porto, and founded on 22 March 1911. It is the second largest Portuguese university by number of enrolled students, after the University of Lisbon, and has one of the most noted research outputs in Portugal.

Property	Value
dbo:abstract	• The University of Porto (Universidade do Porto) is a Portuguese public university located in Porto, and founded on 22 March 1911. It is the second largest Portuguese university by number of enrolled students, after the University of Lisbon, and has one of the most noted research outputs in Portugal. (en)
dbo:affiliation	• dbr:CESAER
dbo:city	• dbr:Porto
dbo:country	• dbr:Portugal
dbo:motto	• Virtus Unita Fortius Agit
dbo:numberOfPostgraduateStudents	• 8235 (xsd:nonNegativeInteger)
dbo:numberOfStudents	• 30640 (xsd:nonNegativeInteger)
dbo:numberOfUndergraduateStudents	• 22405 (xsd:nonNegativeInteger)
dbo:officialSchoolColour	• Gold
dbo:other	• 742 (xsd:integer)
dbo:thumbnail	• wiki-commons:Special:FilePath/Fonte_dos_leões_e_reitoria.jpg?width=300

Example of the "University of Porto" entity in DBpedia
https://dbpedia.org/page/University_of_Porto

```
<http://dbpedia.org/resource/University_of_Porto> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/University> .

<http://dbpedia.org/resource/University_of_Porto> <http://xmlns.com/foaf/0.1/homepage>  <https://sigarra.up.pt/up/en/WEB_PAGE.INICIAL%7C2=www.up.pt> .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/country> <http://dbpedia.org/resource/Portugal> .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/city>  <http://dbpedia.org/resource/Porto> .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/wikiPageUsesTemplate>  <http://dbpedia.org/resource/Template:Color_box> .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/wikiPageWikiLink>  <http://dbpedia.org/resource/Higher_education_in_Portugal> .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/rector>  "António Sousa Pereira"@en .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/motto> "Virtus Unita Fortius Agit" .

<http://dbpedia.org/resource/University_of_Porto> <http://xmlns.com/foaf/0.1/name>  "Universidade do Porto"@en .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/ontology/abstract>  "The University of Porto (Universidade do Porto) is a Portuguese public
university located in Porto, and founded on 22 March 1911. It is the second largest Portuguese university by number of enrolled students, after the University of
Lisbon, and has one of the most noted research outputs in Portugal."@en .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/country> <http://dbpedia.org/resource/Portugal> .

<http://dbpedia.org/resource/University_of_Porto> <http://dbpedia.org/property/nativeName>  "Universidade do Porto"@en .
```

Excerpt from the RDF definition of the "University of Porto" entity in DBpedia in N-Triples
https://dbpedia.org/page/University_of_Porto

Semantic Web

- ➔ The Semantic Web is a term coined be Tim Berners-Lee in 2001 referring to an envisioned extension to the original web.
- ➔ While the original web is a medium of documents for people — a Web of Documents.
- ➔ The semantic web is meant to be a medium for machines (intelligent agents) that have access to data and rules for reasoning about the data — a Web of Data.
- ➔ Semantic Web technologies include: URI, XML, RDF, various serializations of RDF (Turtle, N-Triples, RDFa, etc), SPARQL, OWL, etc — commonly referred to as the Semantic Web Stack.
- ➔ This vision is yet (?) to come true, however it has contributed to the development and publication of structured data in an unprecedented scale.

Entity-Oriented Search

- ➔ Entity-Oriented Search is the search paradigm of organizing and accessing information centered around entities, and their attributed and relationships.
- ➔ From a user perspective, entities are natural units for organizing information. Allowing users to interact with specific entities offers a richer and more effective user experience than what is provided by conventional document-based retrieval systems.
- ➔ From a machine perspective, entities allow for a better understanding of search queries, of document content, and even of users. Entities enable search engines to be more effective.

Entity-Oriented Search

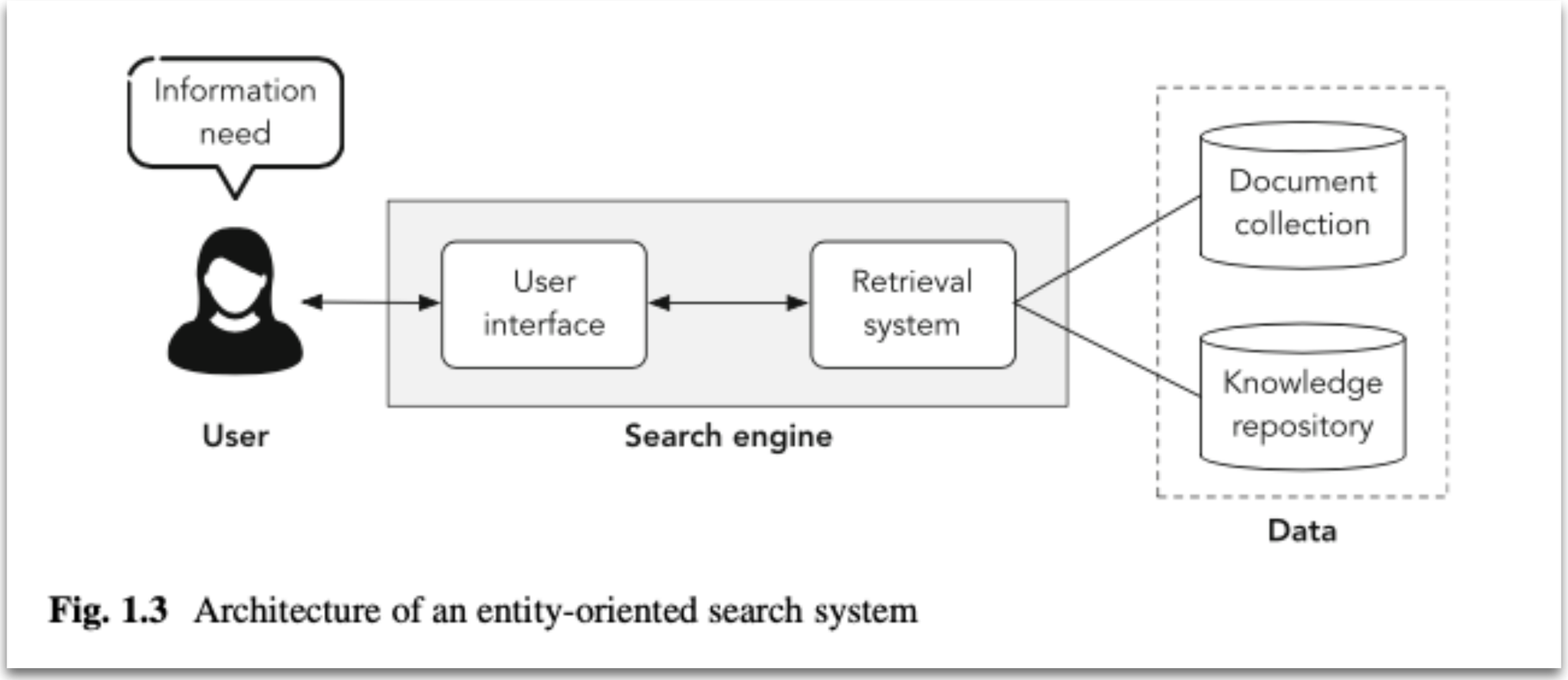


Fig. 1.3 Architecture of an entity-oriented search system

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Users and Information Needs

- ➔ Users articulate their information needs in many different ways.
- ➔ **Keyword queries**, also known as "free text queries", have become the most common approach with the advent of the web. Keyword queries are easy to formulate but imprecise.
- ➔ **Structured queries**, structured data sources are traditionally queried using formal query languages (e.g., SQL, SPARQL). These queries are very precise and usually intended for expert users with well-defined and precise information needs.
- ➔ **Keyword++ queries**, correspond to keyword queries that are complemented with additional structural elements, e.g. [retrieval site:fe.up.pt].
- ➔ **Natural language queries**, are formulated using natural language, the same way one would express it in a conversation, e.g. ["What is the capital of Nepal?"].
- ➔ **Zero-queries**, the traditional information access is reactive — the search system responds to a user-initiated query. Proactive systems, anticipate the user's information need, without requiring the user to issue a query. In practice, the context of the user is used to infer information needs.

Data

➔ Data can be grouped in three types.

➔ **Unstructured data**, which can be found in vast quantities in a variety of forms: web pages, spreadsheets, emails, tweets, etc. All these may be treated as textual documents, i.e. a sequence of words.

➔ **Semi-structured data**, is characterized by the lack of rigid, formal structure. Typically, it contains tags or other markup to separate textual content from semantic elements, e.g. HTML data.

➔ **Structured data**, adheres to a predefined (fixed) schema and is typically organized in a tabular format (e.g., relational databases). The schema defines how the data is organized and imposes constrains to ensure consistency.

Tasks in Entity-Oriented Search

- **Entity Retrieval — entities as the unit of retrieval**, between 40% to 70% of the queries in web search mention or target specific entities. Such queries are better answered by returning a ranked list of entities, as opposed to a list of documents. Challenges involved:
  - 1) how to represent information needs;
  - 2) how to represent entities;
  - 3) how to match those representations.
- **Entity Linking — entities for knowledge representation**, recognizing mentions of entities in text and associating these mentions with the corresponding entities in knowledge bases.
- **Entities for an enhanced user experience**, besides providing meaningful retrieval as information organization units, entities can improve the user experience throughout the entire search process.

Data Sources

Data Types and Search

➔ Recall the basic data types of data sources.

Fig. 2.1 The data spectrum

Table 2.1 Comparison of unstructured, semi-structured, and structured data search

	Unstructured	Semi-structured	Structured
Unit of retrieval	Documents	Objects	Tuples
Schema	No	Self-describing	Fixed
Queries	Keyword	Keyword++	Formal languages

Images from Entity-Oriented Search, Springer, K. Balog (2018)

Knowledge Bases

- A knowledge base is comprised of a large set of assertions about the world.
- When the emphasis is on the relationships between the entities, the term knowledge graph is commonly used.
- Resource Description Framework (RDF) is a language designed to describe "things", which are referred to as resources.
- Each resource is assigned a Uniform Resource Identifier (URI), making it uniquely and globally identifiable.
- Each RDF statement is a triple, consisting of subject, predicate, and object components
  - Subject, always a URI, denoting a resource;
  - Predicate, always a URI, corresponding to a relationship or property of the subject resource;
  - Object, either a URI (referring to another resource) or a literal.

Knowledge Bases

Michael Schumacher (born 3 January 1969) is a retired German racing driver, who raced in Formula One for Ferrari.

<dbr:Michael_Schumacher>	<foaf:name>	"Schumacher, Michael"
<dbr:Michael_Schumacher>	<dbo:birthPlace>	<dbr:West_Germany>
<dbr:Michael_Schumacher>	<dbo:birthDate>	"1969-01-03"
<dbr:Michael_Schumacher>	<rdf:type>	<dbo:RacingDriver>
<dbr:Michael_Schumacher>	<dct:subject>	<dbc:Ferrari_Formula_One_drivers>

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Knowledge Bases

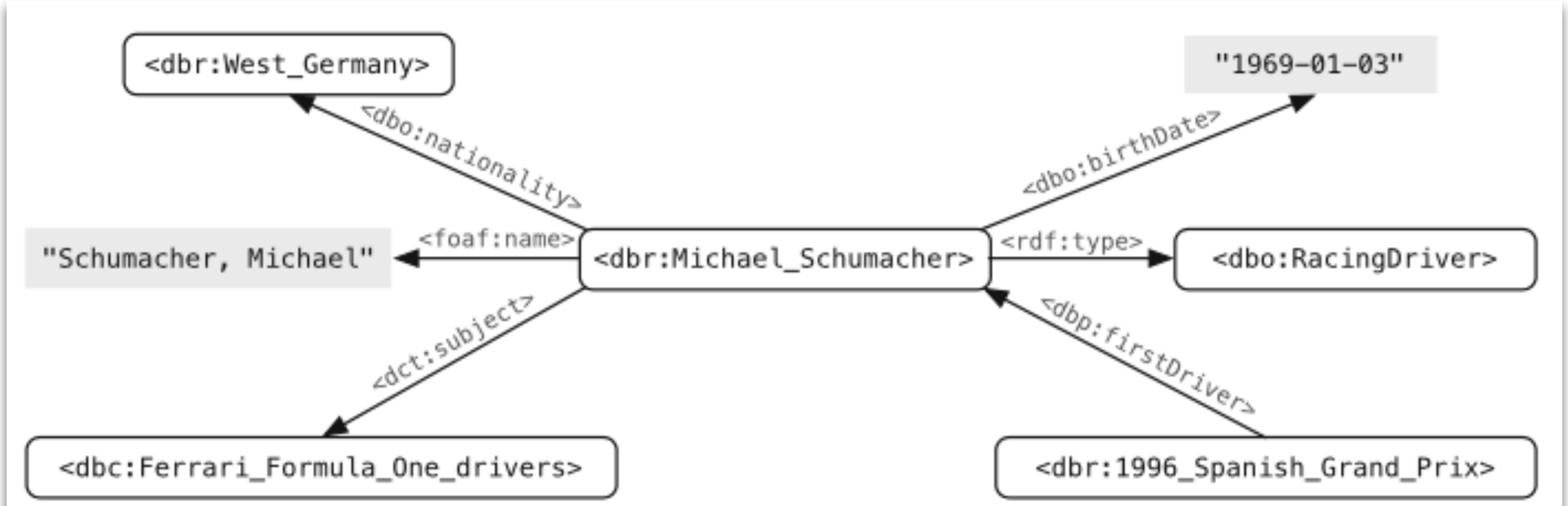


Fig. 2.3 Excerpt of an RDF graph (taken from DBpedia). URIs (i.e., entities) are represented by rounded rectangles, literals (i.e., attribute values) are denoted by shaded rectangles

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Related Technologies

- ➔ RDF describes the instance level in the knowledge base.
- ➔ RDFS and OWL are vocabularies for ontological modeling
  - ➔ An ontology is a means to formalize knowledge. Building blocks of an ontology include classes, instances, relations, attributes, restrictions, and rules and axioms.
- ➔ Serializations for RDF data: Notation-3, Turtle, N-Triples, RDFa, and RDF/JSON.
- ➔ SPARQL is structured query language for retrieving and manipulating RDF data.
- ➔ Triplestores are special-purpose databases designed for storing and querying RDF data.

Public Knowledge Bases

- DBpedia
  - Extracted from Wikipedia (mostly from inboxes) using a set of manually constructed mapping rules.
  - Community effort, users collaboratively create and edit the mapping rules.
  - Available in multiple languages.
  - Contains over 5 million entities (English).

DBpedia

DBpedia Browse using Formats Faceted Browser Sparql Endpoint

About: Porto

An Entity of Type: Municipality, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

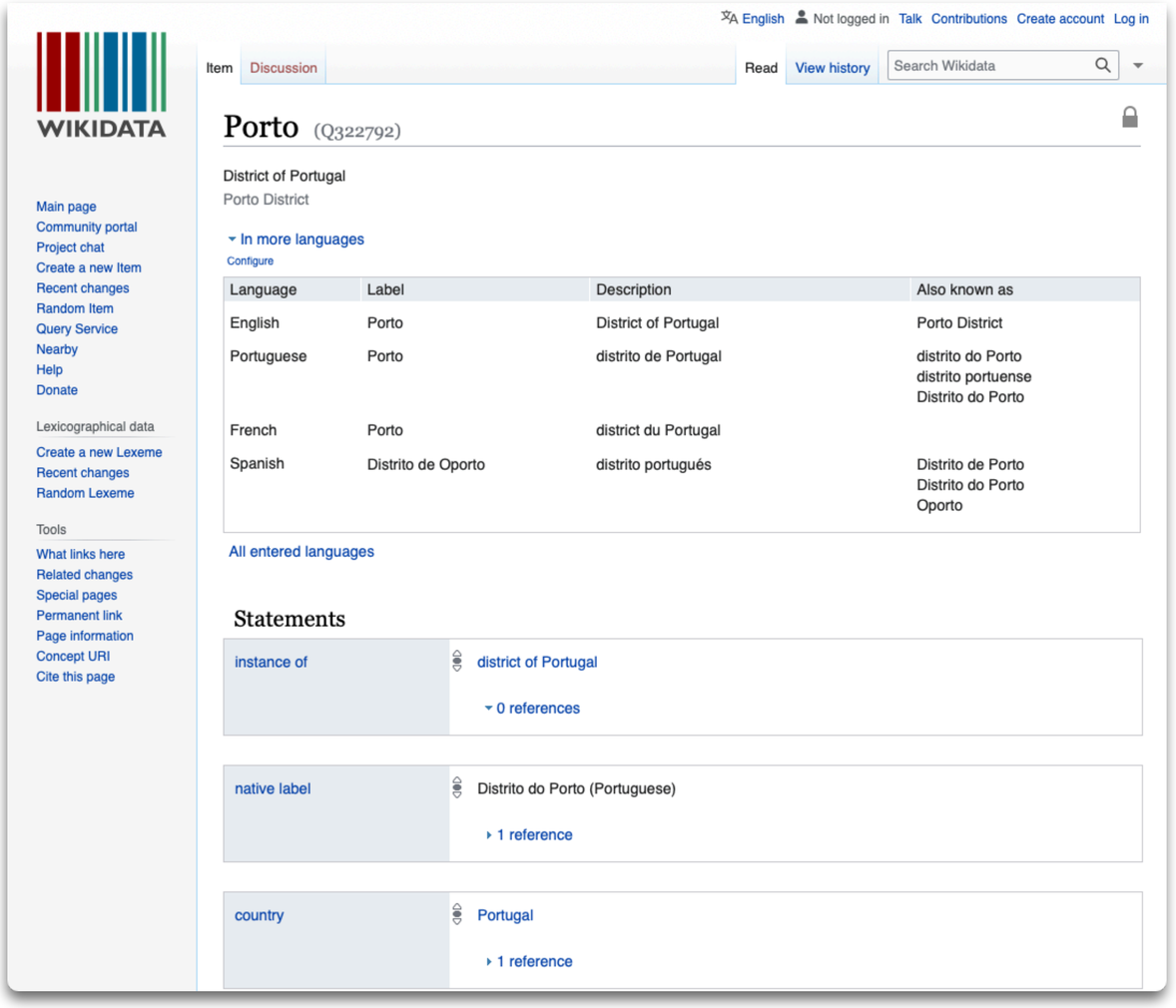
Porto or Oporto (Portuguese pronunciation: ['portu]) is the second-largest city in Portugal and one of the Iberian Peninsula's major urban areas. Porto city proper, which is the entire municipality of Porto, is small compared to its metropolitan area, with an estimated population of just 215,945 people in a municipality with only 41.42 km2. Porto's metropolitan area has around 1.7 million people (2019) in an area of 2,395 km2 (925 sq mi), making it the second-largest urban area in Portugal. It is recognized as a global city with a Gamma + rating from the Globalization and World Cities Research Network.

Property	Value
dbo:PopulatedPlace/areaTotal	• 41.42
dbo:abstract	• Porto or Oporto (Portuguese pronunciation: ['portu]) is the second-largest city in Portugal and one of the Iberian Peninsula's major urban areas. Porto city proper, which is the entire municipality of Porto, is small compared to its metropolitan area, with an estimated population of just 215,945 people in a municipality with only 41.42 km2. Porto's metropolitan area has around 1.7 million people (2019) in an area of 2,395 km2 (925 sq mi), making it the second-largest urban area in Portugal. It is recognized as a global city with a Gamma + rating from the Globalization and World Cities Research Network. Located along the Douro River estuary in northern Portugal, Porto is one of the oldest European centres, and its core was proclaimed a World Heritage Site by UNESCO in 1996, as "Historic Centre of Porto, Luiz I Bridge and Monastery of Serra do Pilar". The historic area is also a National Monument of Portugal. The western part of its urban area extends to the coastline of the Atlantic Ocean. Its settlement dates back many centuries, when it was an outpost of the Roman Empire. Its combined Celtic-Latin name, Portus Cale, has been referred to as the origin of the name Portugal, based on transliteration and oral evolution from Latin. In Portuguese, the name of the city includes a definite article: o Porto ("the port" or "the harbor"), which is where its English name "Oporto" comes from. Port wine, one of Portugal's most famous exports, is named after Porto, since the metropolitan area, and in particular the cellars of Vila Nova de Gaia, were responsible for the packaging, transport, and export of fortified wine. In 2014 and 2017, Porto was elected The Best European Destination by the Best European Destinations Agency. Porto is on the Portuguese Way path of the Camino de Santiago. (en)

Public Knowledge Bases

- Wikidata
  - Operated by the Wikimedia Foundation.
  - Its goal is to provide the same information as Wikipedia, but in a structured format.
  - Wikidata considers "claims" not "facts"
    - Each claim must be supported by a reference;
    - Claims can contradict each other and coexist, thereby allowing opposing views to be expressed (e.g., different political positions).

Wikidata


English
Not logged in
Talk
Contributions
Create account
Log in
WIKIDATA
Item
Discussion
Read
View history
Search Wikidata
Porto (Q322792)
District of Portugal
Porto District
In more languages
Configure
Language
Label
Description
Also known as
English
Porto
District of Portugal
Porto District
Portuguese
Porto
distrito de Portugal
distrito do Porto
distrito portuense
Distrito do Porto
French
Porto
district du Portugal
Spanish
Distrito de Oporto
distrito portugués
Distrito de Porto
Distrito do Porto
Oporto
All entered languages
Statements
instance of
district of Portugal
0 references
native label
Distrito do Porto (Portuguese)
1 reference
country
Portugal
1 reference
Main page
Community portal
Project chat
Create a new Item
Recent changes
Random Item
Query Service
Nearby
Help
Donate
Lexicographical data
Create a new Lexeme
Recent changes
Random Lexeme
Tools
What links here
Related changes
Special pages
Permanent link
Page information
Concept URI
Cite this page

Proprietary Knowledge Bases

- ➔ Google Knowledge Graph
- ➔ Facebook Entity Graph
- ➔ Microsoft Satori

Entity Ranking

Ad Hoc Entity Retrieval Task

➔ Ad hoc entity retrieval is the task of answering queries with a ranked list of entities.

➔ "Ad hoc" refers to the standard form of retrieval in which the user, motivated by an ad hoc (extemporary) information need, initiated the search process by formulating and issuing a query.

Table 3.1 Example entity search queries taken from various benchmarking campaigns [4]

Query
martin luther king
disney orlando
Apollo astronauts who walked on the Moon
Winners of the ACM Athena award
EU countries
Hybrid cars sold in Europe
birds cannot fly
Who developed Skype?
Which films starring Clint Eastwood did he direct himself?

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Main Strategy

- ➔ Model the "entity retrieval" problem as a "document retrieval" problem and take advantage of all the existing body of work.
- ➔ Create an entity description, or "profile document", for each entity in the catalog, containing all the existing "knowledge" about that entity, based on the available data.
- ➔ Once these entity descriptions are created, they can be indexed and ranked using existing document retrieval algorithms.
- ➔ Involves two steps, discussed in the next slides:
  - ➔ Constructing the profile documents;
  - ➔ Ranking the profile documents.

Constructing Entity Descriptions

- ➔ First step is to create a "profile document" with all the information we have about that entity.
- ➔ This will serve as the textual representation of the given entity, the "entity description".
- ➔ Entity descriptions can be assembled by considering the textual content, from a document collection, in which the entities occur.
- ➔ In some cases, such descriptions may already be easily available — e.g., Wikipedia page of an entity. Also, there is a lot of information about entities organized and stored in knowledge bases.
- ➔ Three possible data sources: unstructured, semi-structured and structured.
- ➔ The goal is to estimate a term count associated with each entity.

Entity Components

- ➔ The estimative of the number of times a term is associated with an entity can be used to compute standard components, in an analogous way to the document model.
- ➔ **Entity length**, is the total number of terms in the entity description.
- ➔ **Term frequency (TF)**, is the normalized term count (by length) in the entity description.
- ➔ **Entity frequency (EF)**, is the number of entities in which the term occurs.
- ➔ **Inverse entity frequency (IEF)**, is the log normalized ratio between the total number of entities in the catalog, and the entity frequency.

Entity Representations from Unstructured Data

- ➔ This is the scenario where we want to find entities in arbitrary document collections.
- ➔ An approach is to have documents annotated with the entities that are references in them, and use these documents as a proxy to connect terms and entities.
- ➔ These annotations may be provided by human editors or be automated (e.g., entity linking).
- ➔ In this scenario, the estimated entity term count is obtained by considering each document, and adding the number of co-occurrences between a term and an entity, weighted by the strength of the association between the entity and the document.

Entity Representations from Semi-Structured Data

- A significant portion of the web content is already organized around entities, e.g., Wikipedia page about a person, IMDb page about a movie or an actor.
- The standard way of incorporating the internal document structure into the retrieval model is through the use of document fields, where document segments correspond to fields — e.g., title, summary, introduction, etc.
- The standard statistics are now computed at field-level, e.g. the number of times a term appears in a field of the description of an entity.
- A special "catch-all" field is introduced to
  - (1) quickly filter entities in a first-pass retrieval;
  - (2) since entity description fields are often sparse, improve ranking by combining field-level scores with entity-level (the catch-all field) scores.

Entity Representations from Semi-Structured Data

Fig. 3.4 Web page of the movie THE MATRIX from IMDb (http://www.imdb.com/title/tt0133093/)

Table 3.3 Fielded entity description created for THE MATRIX, corresponding to Fig. 3.4

Field	Content
Name	The Matrix
Genre	Action, Sci-Fi
Synopsis	A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers
Directors	Lana Wachowski (as The Wachowski Brothers), Lilly Wachowski (as The Wachowski Brothers)
Writers	Lilly Wachowski (as The Wachowski Brothers), Lana Wachowski (as The Wachowski Brothers)
Stars	Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss
Catch-all	The Matrix Action, Sci-Fi A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers. Lana Wachowski (as The Wachowski Brothers), Lilly Wachowski (as The Wachowski Brothers) Lilly Wachowski (as The Wachowski Brothers), Lana Wachowski (as The Wachowski Brothers) Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss

Images from Entity-Oriented Search, Springer, K. Balog (2018)

Entity Representations from Structured Data

- There is a growing number of large general-purpose publicly available knowledge bases.
- In knowledge bases information is organized using the RDF data model, specifically
  - each entity is uniquely identified by its URI (Uniform Resource Identifier);
  - its properties are described in the form of subject-predicate-object (SPO) triples.
- To represent an entity from a knowledge base, we consider the immediate vicinity of the entity node — i.e. the SPO tripes where the entity appears either as subject or object.
  - `<http://dbpedia.org/resource/Porto> <http://dbpedia.org/property/name> "Porto"@en .`
  - `<http://dbpedia.org/resource/Porto> <http://xmlns.com/foaf/0.1/nick> "(\"The Undefeated City\"),"@en`
  - `<http://dbpedia.org/resource/Porto> <http://www.w3.org/2002/07/owl#sameAs> <http://www.wikidata.org/entity/Q36433>`
- From the perspective of constructing entity representations, this is conceptually no different from having data stored in relational databases, where the same information is available through fields and foreign-key relationships.

Entity Representations from Triples

- ➔ The number of potential fields is huge (in the 1000s), thus we need a different strategy.
- ➔ A common solution is predicate folding — grouping predicates together into a small set of predefined categories, so that we have a handful go fields.
- ➔ Predicates may be grouped based on their type or importance.
- ➔ Common entity fields: name, name variants, attributed, types, outgoing relations, incoming relations, top predicates, catch-all.

RDF Graph

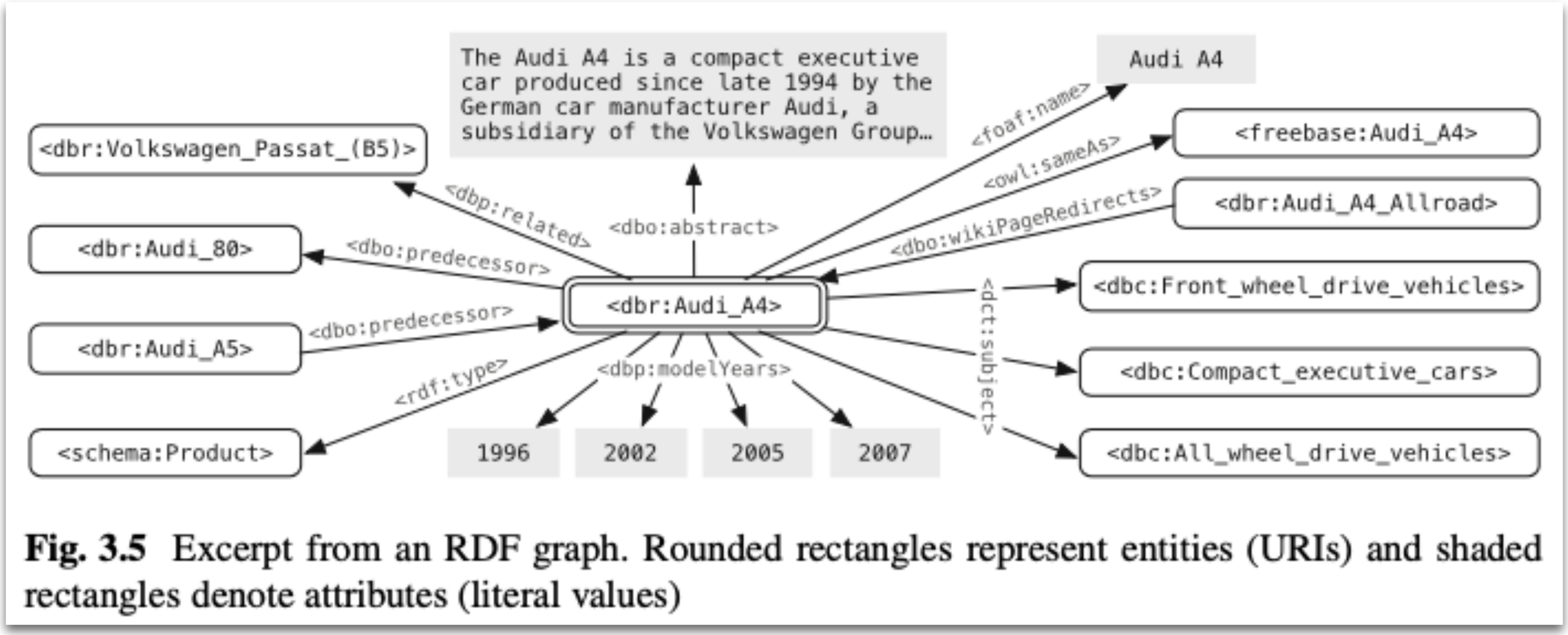


Fig. 3.5 Excerpt from an RDF graph. Rounded rectangles represent entities (URIs) and shaded rectangles denote attributes (literal values)

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Fielded Entity Description

Table 3.4 Excerpt from the fielded entity description of AUDI A4, corresponding to Fig. 3.5

Field	Content
Name	Audi A4
Name variants	Audi A4 … Audi A4 Allroad
Attributes	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [. . .] . . . 1996 . . . 2002 . . . 2005 . . . 2007
Types	Product . . . Front wheel drive vehicles . . . Compact executive cars . . . All wheel drive vehicles
Outgoing relations	Volkswagen Passat (B5) . . . Audi 80
Incoming relations	Audi A5
`<foaf:name>`	Audi A4
`<dbo:abstract>`	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [. . .]
Catch-all	Audi A4 . . . Audi A4 . . . Audi A4 Allroad . . . The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group [. . .] . . . 1996 . . . 2002 . . . 2005 . . . 2007 . . . Product . . . Front wheel drive vehicles . . . Compact executive cars . . . All wheel drive vehicles . . . Volkswagen Passat (B5) . . . Audi 80 . . . Audi A5

Table from Entity-Oriented Search, Springer, K. Balog (2018)

Commonly Used Fields (1)

➔ **Name** contains the name(s) of the entity.

  ➔ The two main predicates mapped to this field are <foaf:name> and <rdfs:label>.

  ➔ One might follow a simple heuristic and additionally consider all predicates ending with "name", "label", or "title".

➔ **Name variants** (aliases) may be aggregated in a separate field.

  ➔ In DBpedia, such variants may be collected via Wikipedia redirects (via <dbo:wikiPageRedirects>) and disambiguations (using <dbo:wikiPageDisambiguates>).

➔ **Attributes** includes all objects with literal values, except the ones already included in the name field.

  ➔ In some cases, the name of the predicate may also be included along with the value, e.g., "founding date 1964" (vs. just the value part, "1964").

Commonly Used Fields (2)

- **Types** holds all types (categories, classes, etc.) to which the entity is assigned.
  - Commonly, <rdf:type> is used for types;
  - In DBpedia, <dct:subject> is used for assigning Wikipedia categories, which may also be considered as entity types.
- **Outgoing relations** contains all URI objects, i.e., names of entities (or resources in general) that the subject entity links to.
  - If the types or name variants fields are used, then those predicates are excluded;
  - Values might be prefixed with the predicate name, e.g., "spouse Michelle Obama".
- **Incoming relations** is made up of subject URIs from all SPO triples where the entity appears as object.
- **Top predicates** may be considered as individual fields
  - e.g., include the top-1000 most frequent DBpedia predicates as fields.
- **Catch-all** (or content) is a field that amasses all textual content related to the entity.

Triples to Text

- How to represent triples as text?
- Triple objects values are either URIs (links) or literals.
- Literals can be treated as regular text.
- URIs need to be converted to human-readable
  - Some are user-friendly: https://dbpedia.org/page/Porto
  - Others are not: http://www.wikidata.org/entity/Q36433
- URI resolution is the process of finding the a human-readable name/label for a URI.

URI Resolution

- The specific predicate that holds the name of a resource depends on the RDF vocabulary used.
- Commonly, <foaf:name> or <rdfs:label> are used
- Given a SPO triple, for example
  - `<dbr:Audi_A4>  <rdf:type>  <dbo:MeanOfTransportation>`
- The corresponding resources's name is contained in the object element of this triple:
  - `<dbo:MeanOfTransportation>  <rdfs:label>  “mean of transportation"`

Ranking Entities

➔ With the term-based entity representations, we now need to rank entities with respect to their relevance to a search query.

➔ This can be viewed as a the problem of assigning a score to each entity in the catalog.

➔ The retrieval models from document ranking — e.g. vector space model, language models, probabilistic models — are used for entity scoring by replacing the *document* with *entity* in the equations.

Entity Ranking Evaluation

- Evaluation of ad hoc entity retrieval is analogous to that of ad hoc document retrieval.
- Given a ranked list of items, the relevance of each item is judged with respect to the underlying information need.
  - Set-based measures, e.g. precision, recall, precision at rank cutoff k (P@10, P@20).
  - Rank-based measures, e.g. average precision (AvP), mean average precision (MAP).
- The use of text collections is the de facto standard standard for evaluation in IR. Some relevant collections that can be used for entity retrieval are: INEX Entity Ranking (XER), TREC Entity, Semantic Search Challenge, INEX Linked Data.

Entity Ranking Summary

- ➔ In this section we discussed approaches for ranking entities in various datasets, from unstructured documents, to structured knowledge bases.
- ➔ Most of the effort in this area has been in constructing term-based representations of entities, which can be ranked using traditional document retrieval techniques.
- ➔ Unstructured entity representations with a bag-of-words retrieval models usually provide solid performance and a good starting point.
- ➔ Other approaches exist, most notably the use of specific characteristics of entities, such as relationships.

Entity Linking

Entity Linking

➔ Entity linking is the task of recognizing entity mentions in text and linking them to the corresponding entries in a knowledge base.

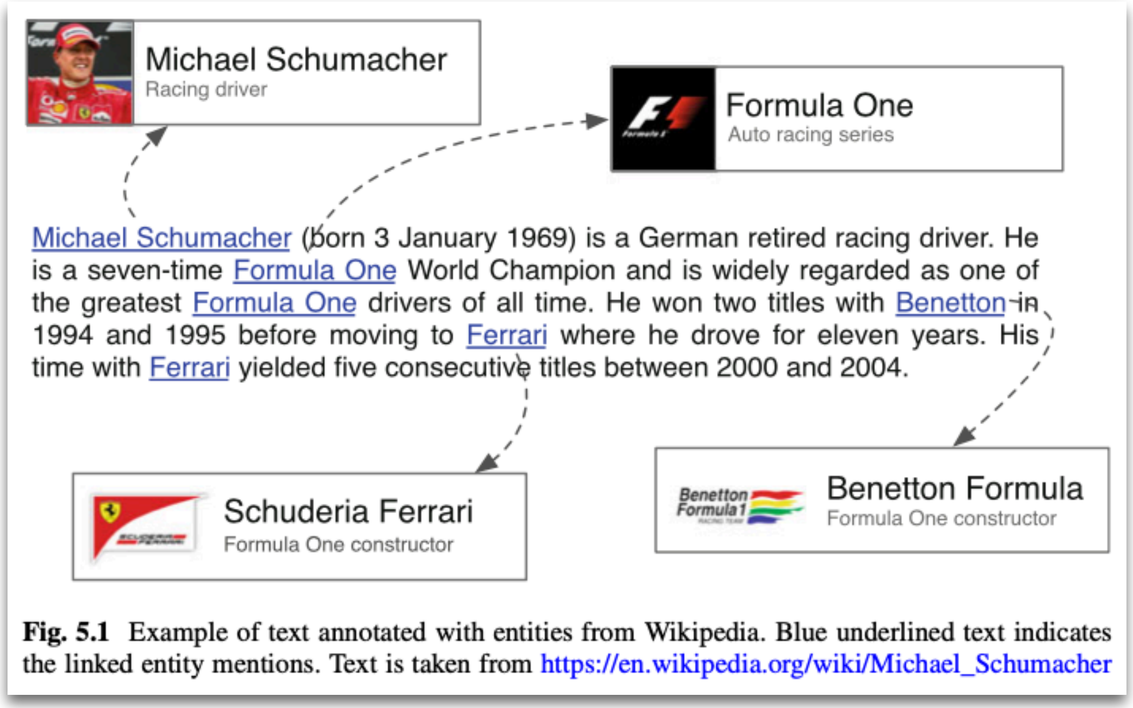
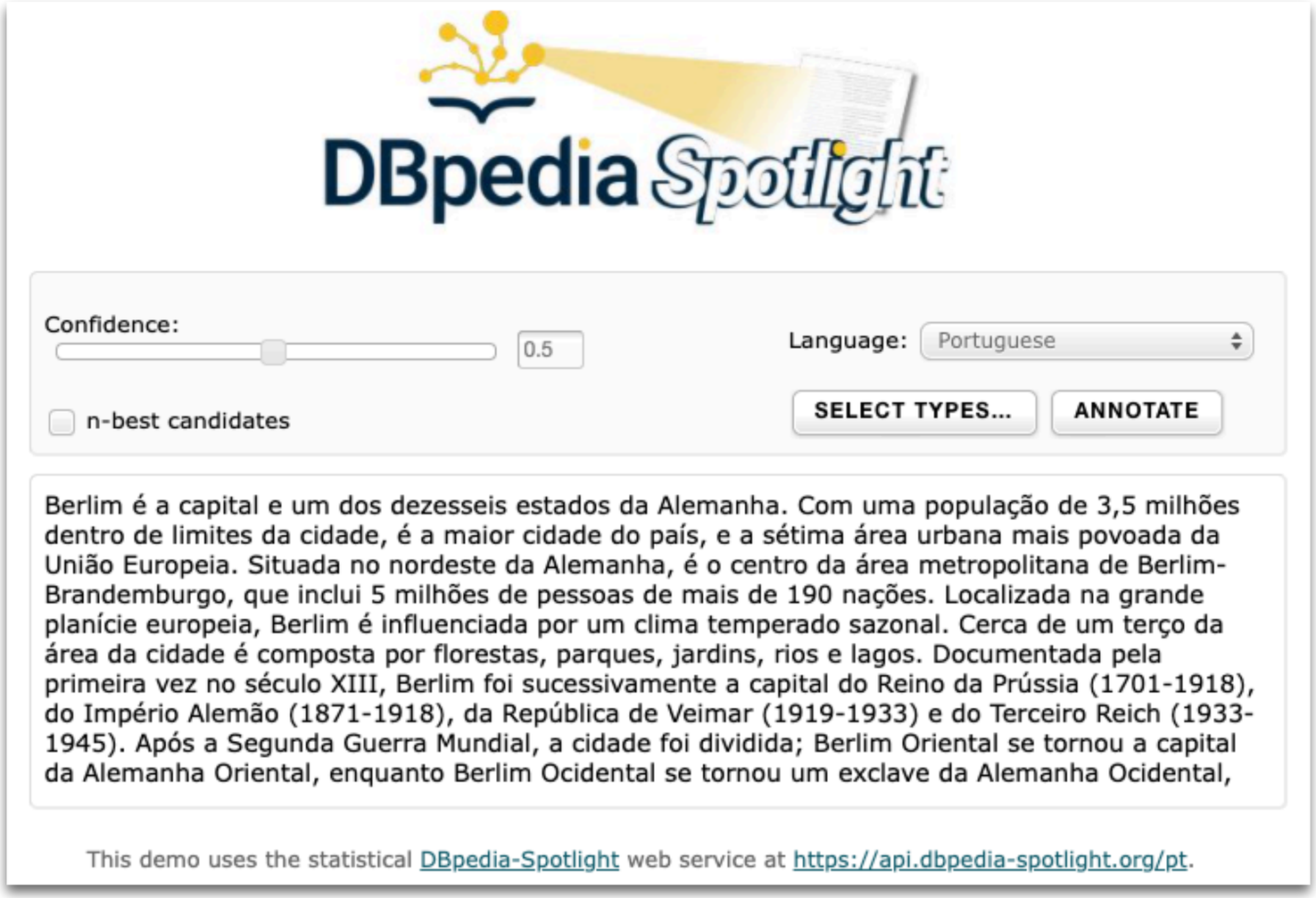


Fig. 5.1 Example of text annotated with entities from Wikipedia. Blue underlined text indicates the linked entity mentions. Text is taken from https://en.wikipedia.org/wiki/Michael_Schumacher

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Practical Example (Text)



Available online at: https://demo.dbpedia-spotlight.org/

Practical Example (Annotated)

Available online at: https://demo.dbpedia-spotlight.org/

Anatomy of an Entity Linking System

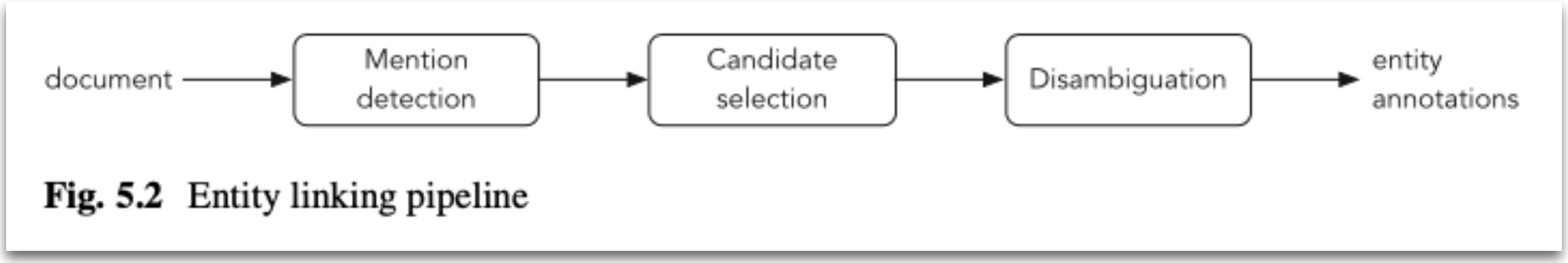


Fig. 5.2 Entity linking pipeline

➔ **Mention detection**, identification of text snippets that can potentially be linked to entities.

➔ **Candidate selection**, generating a set of candidate entities for each mention.

➔ **Disambiguation**, selecting a single entity (or none) for each mention, based on the context.

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Common Approach for Mention Detection

➔ Build a dictionary o entity surface forms, i.e. entities with all names variants.

➔ Check all document n-grams against the dictionary

➔ Filter out undesired entities

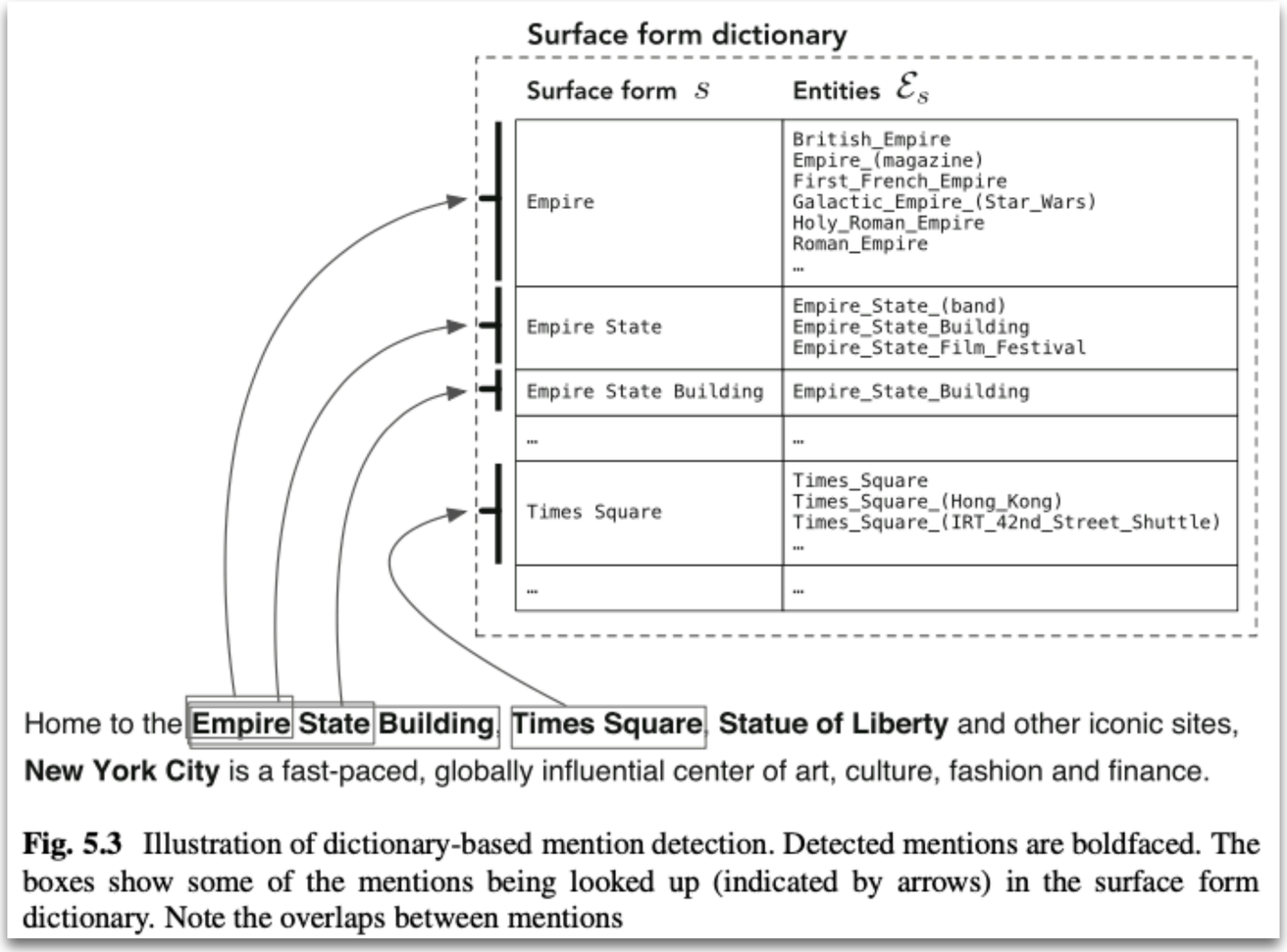


Fig. 5.3 Illustration of dictionary-based mention detection. Detected mentions are boldfaced. The boxes show some of the mentions being looked up (indicated by arrows) in the surface form dictionary. Note the overlaps between mentions

Image from Entity-Oriented Search, Springer, K. Balog (2018)

Entity Linking Systems

Table 5.5 Overview of publicly available entity linking systems

System	Reference KR	Online demo	Web API	Source code
AIDA[a]	YAGO2	Yes	Yes	Yes (Java)
DBpedia Spotlight[b]	DBpedia	Yes	Yes	Yes (Java)
Illinois Wikifier[c]	Wikipedia	No	No	Yes (Java)
TAGME[d]	Wikipedia	Yes	Yes	Yes (Java)
Wikipedia Miner[e]	Wikipedia	No	No	Yes (Java)

➔ http://www.mpi-inf.mpg.de/yago-naga/aida/

➔ http://spotlight.dbpedia.org/

➔ http://cogcomp.cs.illinois.edu/page/download_view/Wikifier

➔ https://tagme.d4science.org/tagme/

➔ https://github.com/dnmilne/wikipediaminer

Table from Entity-Oriented Search, Springer, K. Balog (2018)

References and Further Reading

- K. Balog. Entity-Oriented Search. Springer, 2018.
  https://eos-book.org/

- H. Bast, B. Buchhold, and E. Haussmann. Semantic search on text and knowledge bases. Foundations and Trends in Information Retrieval, 10(2–3):119–271, 2016.