Knowledge Bases and Entity Retrieval

[DAT640] Information Retrieval and Text Mining

Ivica Kostric & Krisztian Balog
University of Stavanger

September 12, 2022

In this module

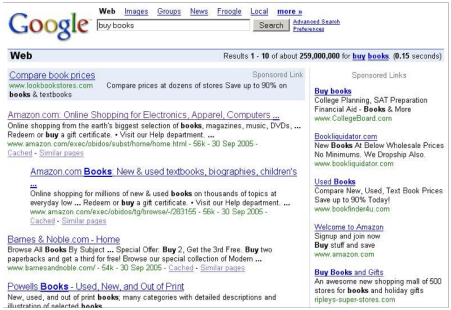
Entity-oriented search

Question

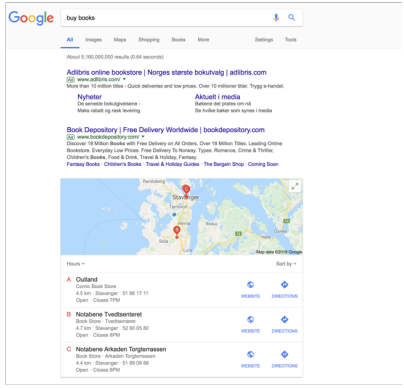
What do you think *semantic search* means?

Examples of semantic search

Query: buy books



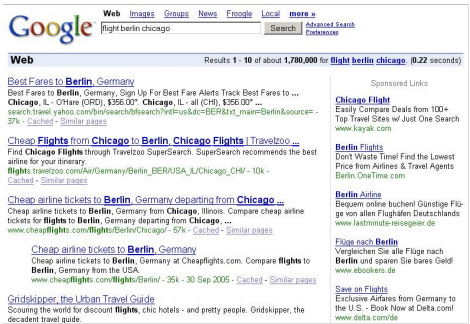
Google result page in 2005

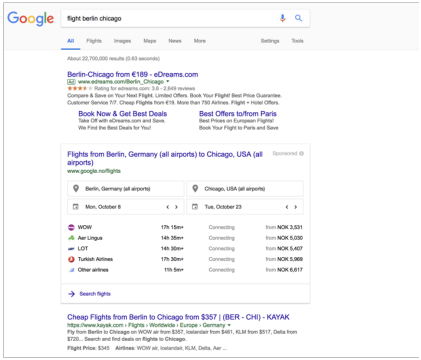


Google result page today

Examples of semantic search

Query: flight berlin chicago





Google result page in 2005

Google result page today

Semantic search

A broad view on semantic search:

Definition

Semantic search encompasses a variety of methods and approaches aimed at aiding users in their information access and consumption activities, by understanding their context and intent.

- "Search with meaning"
  - Beyond literal matches
  - Understanding what the query actually means
  - Searching for *things* instead of *strings*
- Our notion of semantics: references to meaningful, i.e., machine understandable structures

Entity-oriented search

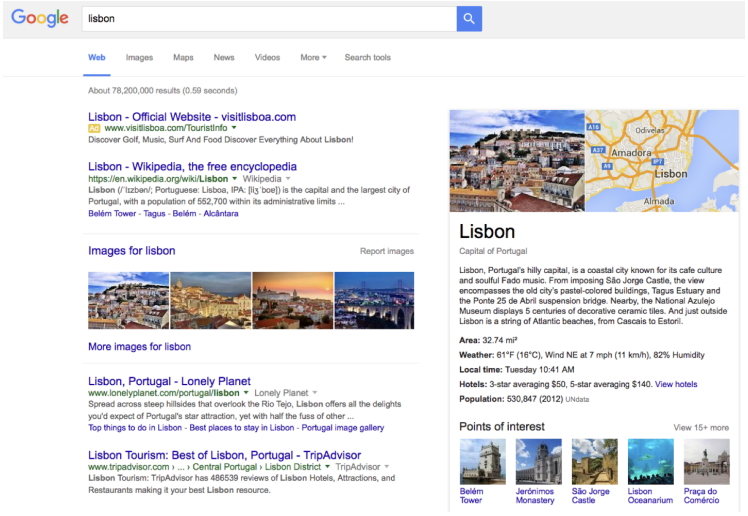
Entity-oriented search to refer to a broad range of information access tasks where entities are used as information objects, instead of or in addition to documents

Definition

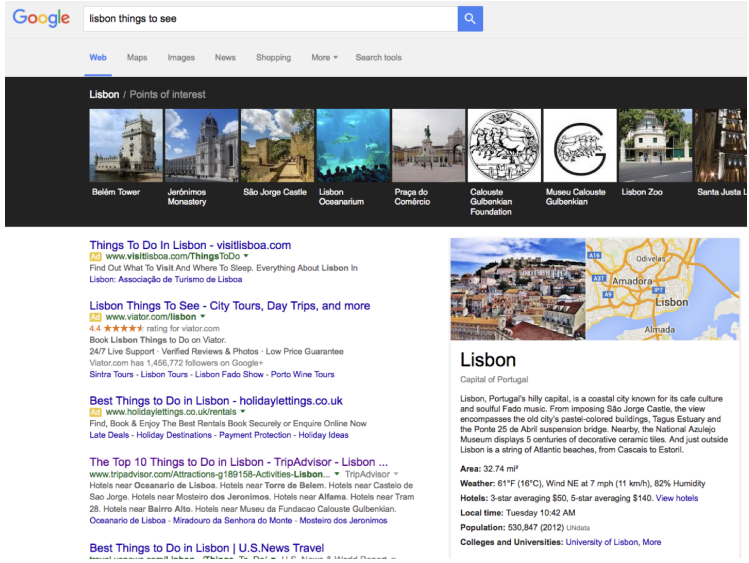
Entity-oriented search is the search paradigm of organizing and accessing information centered around entities, and their attributes and relationships.

- Note: entity-oriented search is a subset of semantic search

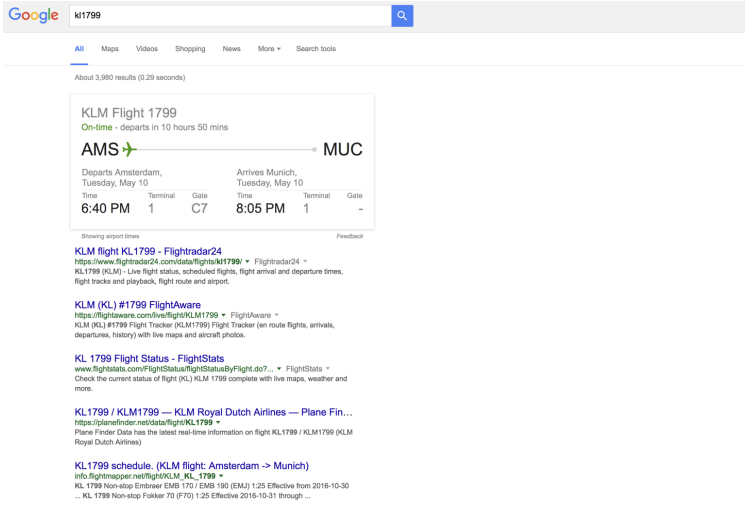
Examples of entity-oriented search



Examples of entity-oriented search



Examples of entity-oriented search



Why?

- From a **user perspective**
  - Entities are natural units for organizing information
    - We care about and mostly think in terms of real-world things and their connections
- From a **machine perspective**
  - Entities allow for a better understanding of search queries, of document content, and even of users (e.g., their context and preferences)
  - Entities enable search engines to be more intelligent

Entity-oriented search

What is an entity?

people

organizations

locations

products

What is an entity?

Commonly accepted definition:

Definition

An *entity* is an object or concept in the real world that can be distinctly identified.

- Issues
  - What does the "real world" mean? (Is "Superman" an entity or not?)
  - Answering this will likely lead to a long philosophical discussion about "existence"

What is an entity?

Pragmatic, data-oriented definition:

Definition

An *entity* is a uniquely identifiable object or thing, characterized by its name(s), type(s), attributes, and relationships to other entities.

Our universe is restricted to some particular registry of entities:

Definition

An *entity catalog* is a collection of entries, where each entry is identified by a unique ID and contains the name(s) of the corresponding entity.

Named entities vs. concepts

- Two main classes of entities may be distinguished
  - *Named entities* are real-world objects that can be denoted by a proper noun
    - For example, specific persons, locations, organizations, products, events, etc.
  - *Concepts* are abstract objects, including, but not limited to
    - Mathematical and philosophical concepts (e.g., “distance,” “axiom,” “quantity”)
    - Physical concepts and natural phenomena (e.g., “gravity,” “force,” “wind”)
    - Psychological concepts (e.g., “emotion,” “thought,” “identity”), and social concepts (e.g., “authority,” “human rights,” “peace”)
- This distinction is mostly of a philosophical nature. From a technical perspective, the exact same methods may be used for names entities and concepts.

Properties of entities

- **Unique identifier**
  - There must be a one-to-one correspondence between each entity identifier (ID) and the (real-world or fictional) object it represents
  - For example, social security number, product EAN, MAC address, etc.
- **Name(s)**
  - Names do not uniquely identify entities; multiple entities may share the same name
  - The same entity may be known by more than a single name (e.g., "Barack Obama," "President Obama")
  - Alternative names are called *surface forms* or *aliases*
- **Type(s)**
  - Entities may be categorized into multiple *entity types*
  - Types can also be thought of as containers (semantic categories) that group together entities with similar properties
  - Analogy to object-oriented programming: an entity of a type is like an instance of a class
  - Entity types are often organized in a hierarchical structure (*type taxonomy*)

Properties of entities (2)

- **Attributes**
  - Entities are characterized by attributes
  - Different types of entities typically have different sets of attributes
    - People: date and place of birth, weight, height, parents, spouses, etc.
    - Places: latitude, longitude, population, postal code(s), country, continent, etc.
  - Attributes always have *literal values*
- **Relationships**
  - May be seen as "typed links" between entities (or attributes where the value is another entity)
  - For example, parents of a person, capital of a country, manufacturer of a product, etc.

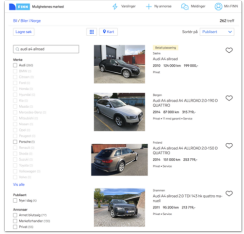
Entity-oriented search

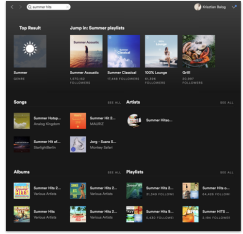
Entity-oriented search tasks

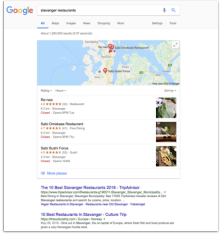
- Entities as the unit of retrieval
  - Entity retrieval
- Entities for knowledge representation
  - Entity linking
  - Knowledge base population
- Entities for an enhanced user experience
  - Query assistance
  - Recommendations

Entity retrieval

- Task: given a search query, return a ranked list of entities (instead of documents)

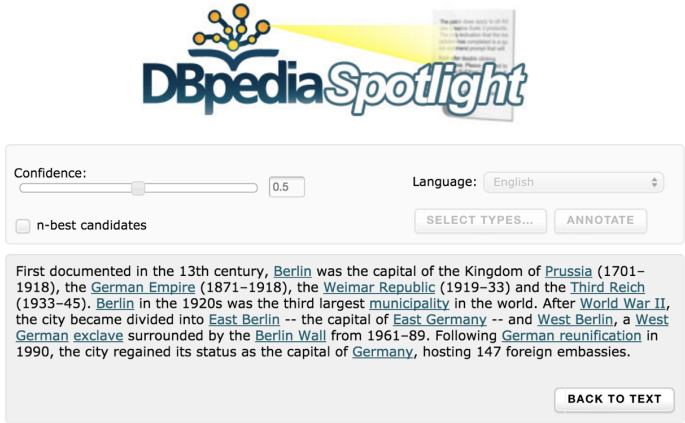






Entity linking

- Task: recognize mentions of entities in text and assign to these unique identifiers from a knowledge repository



Architecture of an entity-oriented search system

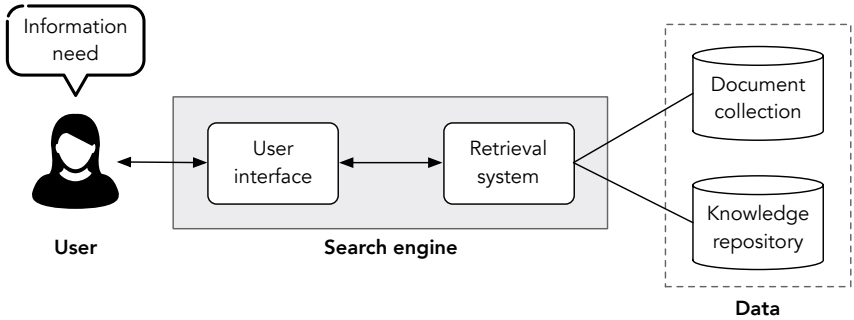


Figure: Illustration is taken from (Balog, 2018) [Fig. 1.3]

Representing properties of entities

Information about entities can be represented and stored in semi-structured or in structured form

Definition

A *knowledge repository* (KR) is a catalog of entities that contains entity type information, and (optionally) descriptions or properties of entities, in a semi-structured or structured format.

- Classic example: Wikipedia
  - Each article in Wikipedia is an entry that describes a particular entity
  - Articles are also assigned to categories (which can be seen as entity types)
  - Wikipedia articles also contain information about attributes and relationships of entities, but not in a structured form

Representing properties of entities

To organize and store information about entities in a structured form, entities may be represented as a set of statements (facts or assertions)

Definition

A *knowledge base* (KB) is a structured knowledge repository that contains a set of facts (assertions) about entities.

- Note: all knowledge bases are also knowledge repositories, but the reverse is not true
- Conceptually, entities in a knowledge base may be seen as nodes of a graph, with the relationships between them as (labeled) edges
  - When this graph nature is emphasized, a knowledge base may also be referred to as a *knowledge graph* (KG)

Representing properties of entities

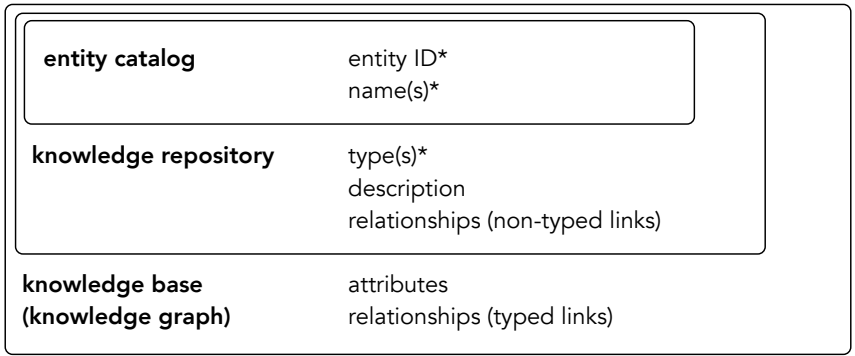


Figure: Illustration is taken from (Balog, 2018) [Fig. 1.2]

Entity-oriented search

Question

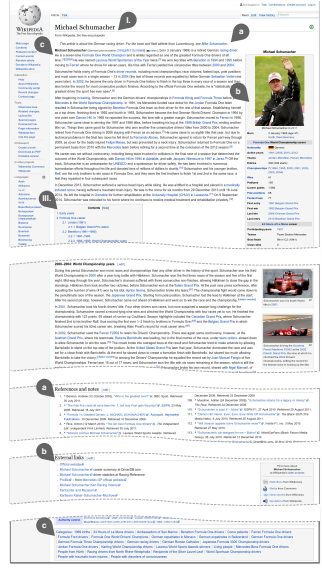
Why is Wikipedia relevant for entity-oriented search?

Wikipedia

- One of the most popular web sites in the world and a trusted source of information for many people
- Content is created through the collaborative effort of a community of users, facilitated by a *wiki* platform
- Available in nearly 300 languages, although English is by far the most popular, with over five million articles

Wikipedia as a knowledge repository

- Most of Wikipedia's entries can be considered as (semi-structured) representations of entities



The anatomy of a Wikipedia article

- Title
- Lead section
  - Disambiguation links
  - Infobox
  - Introductory text
- Table of contents
- Body content
- Appendices and bottom matter
  - References and notes
  - External links
  - Categories

Entity-oriented search

Knowledge base

- A data repository for storing entities and their properties in structured format
- A set of assertions about the world, describing specific entities and their relationships
- Conceptually, it forms a graph (“knowledge graph”)

Resource Description Framework (RDF)

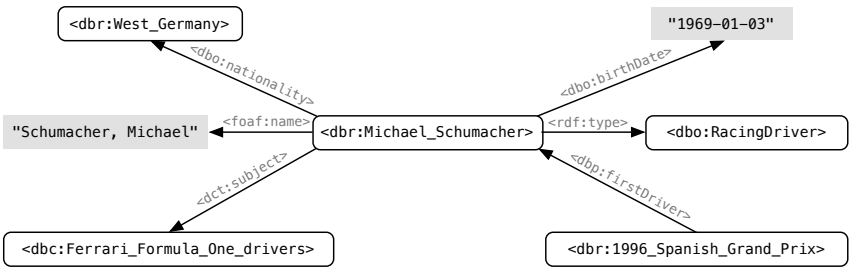
- A language designed to describe "things" (which are referred to as *resources*)
- Each resource is assigned a *Uniform Resource Identifier* (URI), making it uniquely and globally identifiable
- Each RDF statement is a triple, consisting of subject, predicate, and object components
  - **Subject**: 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

Example

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

subject	predicate	object
<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>

Example


<dbr:West_Germany>
<dbo:nationality>
"1969-01-03"
<dbo:birthDate>
"Schumacher, Michael"
<foaf:name>
<dbr:Michael_Schumacher>
<rdf:type>
<dbo:RacingDriver>
<dct:subject>
<dbp:firstDriver>
<dbc:Ferrari_Formula_One_drivers>
<dbr:1996_Spanish_Grand_Prix>

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 formalizing 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 a structured query language for retrieving and manipulating RDF data
- Triplestores are special-purpose databases designed for storing and querying RDF data

Public knowledge bases

- Cyc
  - Started in 1984 with the goal to manually build a knowledge base of everyday common knowledge
  - ... still building and far from complete
  - “one of the most controversial endeavors of the artificial intelligence history”

Public knowledge bases

- DBpedia
  - Extracted from Wikipedia (mostly from infoboxes) 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)

Example

About: Stavanger

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

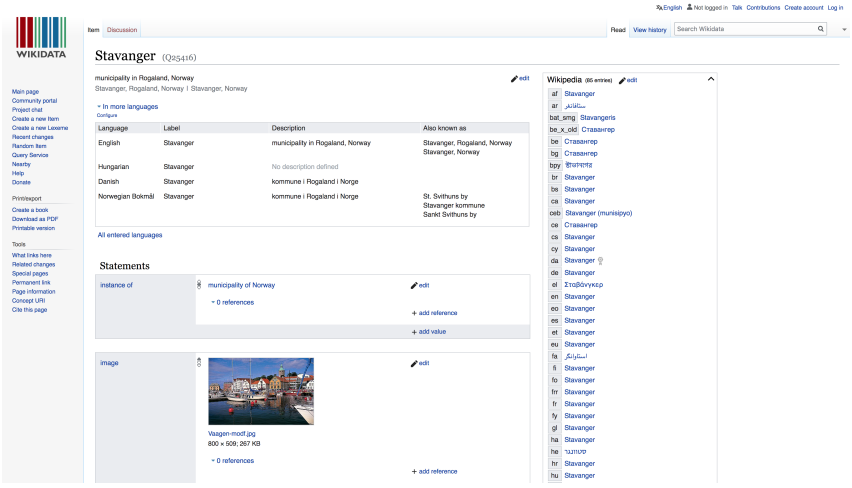
Stavanger /stəˈvæŋər/ (Norwegian pronunciation: [stɑˈvɑŋər]) is a city and municipality in Norway. The city is the third-largest urban zone and metropolitan area in Norway (through conurbation with neighbouring Sandnes) and the administrative centre of Rogaland county. The municipality is the fourth most populous in Norway. Located on the Stavanger Peninsula in Southwest Norway, Stavanger counts its official founding year as 1125, the year the Stavanger Cathedral was completed. Stavanger's core is to a large degree 18th- and 19th-century wooden houses that are protected and considered part of the city's cultural heritage. This has caused the town centre and inner city to retain a small-town character with an unusually high ratio of detached houses, and has contributed signif

Property	Value
dbo:PopulatedPlace/areaMetro	▪ 2598.0
dbo:PopulatedPlace/areaTotal	▪ 71.0
dbo:PopulatedPlace/areaUrban	▪ 77.98
dbo:abstract	▪ Stavanger /stəˈvæŋər/ (Norwegian pronunciation: [stɑˈvɑŋər]) is a city and municipality in Norway. The city is the third-largest urban zone and metropolitan area in Norway (through conurbation with neighbouring Sandnes) and the administrative centre of Rogaland county. The municipality is the fourth most populous in Norway. Located on the Stavanger Peninsula in Southwest Norway, Stavanger counts its official founding year as 1125, the year the Stavanger Cathedral was completed. Stavanger's core is to a large degree 18th- and 19th-century wooden houses that are protected and considered part of the city's cultural heritage. This has caused the town centre and inner city to retain a small-town character with an unusually high ratio of detached houses, and has contributed significantly to spreading the city's population growth to outlying parts of Greater Stavanger. The city's rapid population growth in the late 20th century was primarily a result of Norway's booming offshore oil industry. Today the oil industry is a key industry in the Stavanger region and the city is widely referred to as the Oil Capital of Norway. The largest company in the Nordic region, Norwegian energy company Statoil is headquartered in Stavanger. Multiple educational institutions for higher education are located in Stavanger. The largest of these is the University of Stavanger. Domestic and international military installations are located in Stavanger, among these is the North Atlantic Treaty Organisation's Joint Warfare Center. Other international establishments, and especially local branches of foreign oil and gas companies, contribute further to a significant foreign population in the city. Immigrants make up 11.3% of Stavanger's population. Stavanger has since the early 2000s consistently had an unemployment rate significantly lower than the Norwegian and European average. In 2011, the unemployment rate was less than 2%. The city is also among those that frequent various lists of expensive cities in the world, and Stavanger has even been ranked as the world's most expensive city by certain indexes. Stavanger is served by international airport Stavanger Airport, Sola, which offers flights to cities in most major European countries, as well as a limited number of intercontinental charter flights. The airport was named most punctual European regional airport by flightstats.com in 2010. Every two years, Stavanger organizes the Offshore Northern Seas (ONS), which is the second largest exhibition and conference for the energy sector. Gladmat food festival is also held each year and is considered to be one of Scandinavia's leading food festivals. The city is also known for being one of the nation's premier culinary clusters. Stavanger 2008 European Capital of Culture. (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)

Example


English Not logged in Talk Contributions Create account Log in
WIKIDATA
Item Discussion
Read View history Search Wikidata
Stavanger (Q25416)
municipality in Rogaland, Norway
Stavanger, Rogaland, Norway | Stavanger, Norway
edit
In more languages
Configure
Language Label Description Also known as
English Stavanger municipality in Rogaland, Norway Stavanger, Rogaland, Norway
Stavanger, Norway
Hungarian Stavanger No description defined
Danish Stavanger kommune i Rogaland i Norge
Norwegian Bokmål Stavanger kommune i Rogaland i Norge St. Svithuns by
Stavanger kommune
Sankt Svithuns by
All entered languages
Wikipedia (86 entries) edit
af Stavanger
ar ستافنگر
bat_smg Stavangeris
be_x_old Ставангер
be Ставангер
bg Ставангер
bpy ꯁ্তাভাংগের
br Stavanger
bs Stavanger
ca Stavanger
ceb Stavanger (munisipyo)
ce Ставангер
cs Stavanger
cy Stavanger
da Stavanger
de Stavanger
el Στοβάνγκερ
en Stavanger
eo Stavanger
es Stavanger
et Stavanger
eu Stavanger
fa استاوانگر
fi Stavanger
fo Stavanger
fr Stavanger
fy Stavanger
gl Stavanger
ha Stavanger
he סטוונגר
hr Stavanger
hu Stavanger
Statements
instance of municipality of Norway edit
0 references
add reference
add value
image
Vaagen-mod.jpg
800 × 509; 267 KB
edit
0 references
add reference

Proprietary knowledge bases

- Google Knowledge Graph
  - "... the knowledge graph is one of Google's biggest search milestones of the last decade..."—Amit Singhal, Google's director of search
- Facebook Entity Graph
- Microsoft Satori
- ...

Connecting knowledge bases

- The same entity may be present in multiple knowledge bases
- A special predicate `<owl:sameAs>` can be used to connect URIs across different knowledge bases

subject	predicate	object
`<dbr:Michael_Schumacher>`	`<owl:sameAs>`	`<fb:m.053w4>`
`<dbr:Michael_Schumacher>`	`<owl:sameAs>`	`<<wikidata:Q9671>`

The Web of Data

- Increasingly more data is being exposed on the Web in the form of semantic annotations
  - Microdata, RDFa, JSON-LD
- Strong incentive for websites for marking up their content with semantic metadata: It allows search engines to better understand their content
- Standardization: development of `schema.org`
  - A common vocabulary used by major search providers (including Google, Microsoft, and Yandex) for describing commonly used entity types (including people, organizations, events, products, books, movies, recipes, etc.)

Example

Easy Chicken Satay Recipe - Allrecipes.com

allrecipes.com/recipe/132929/easy-chicken-satay/

★★★★½ Rating: 4,7 - 282 reviews - 2 hrs 45 mins - 418 cal

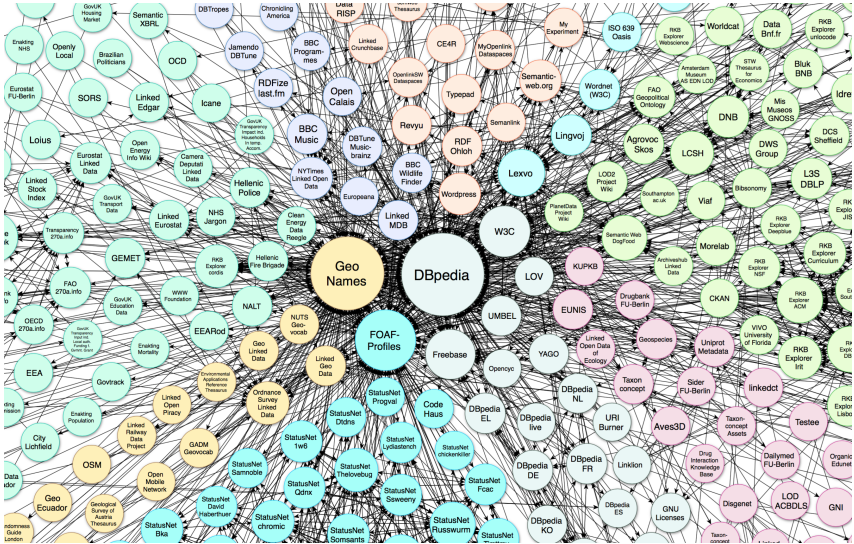
Bring 1 cup coconut milk, 1 tablespoon curry powder, peanut butter, **chicken** stock, and 1/4 cup brown sugar to a simmer in a saucepan over medium-high heat. Simmer for 5 minutes, stirring constantly, until smooth and thickened. Remove from heat and stir in lime juice and soy sauce; season to taste with salt.

```
<section class="ar_recipe_index full-page" itemscope
         itemtype="http://schema.org/Recipe">
    <link href="http://allrecipes.com/recipe/132929/easy-chicken-satay/"
          itemprop="url" />
    <meta itemprop="mainEntityOfPage" content="True" />
```

The Semantic Web

- Historically, data made available in RDF format was referred to as Semantic Web data
- One of the founding principles behind the Semantic Web is that data should be interlinked
- The term *Linked Data (LD)* refers to a set of best practices for publishing structured data on the Web
  - This is facilitated by the special "same-as" predicate
  - A knowledge base published using LD principles should be called *Linked Dataset*
- These "same-as" links connect all Linked Data into a single global data graph
- *Linked Open Data (LOD)* (a casual synonym for the Web of Data) emphasizes the fact that Linked Data is released under an open license

Linked Open Data



Elasticsearch

Elasticsearch

- An open source search engine built on top of Apace Lucene
- It is distributed - indices can be divided into shards and each shard can have zero or more replicas
- All of its functionality is available through a RESTful API

Exercise

E6-0 Elasticsearch Basics

Exercise

E6-1 Term statistics

Entity retrieval

Ad hoc entity retrieval

Entity retrieval is the task of answering queries with a ranked list of entities[1]

Definition

Given a keyword query q and an entity catalog $\mathcal{E}$, *ad hoc entity retrieval* is the task of returning a ranked list of entities $\langle e_1, \ldots, e_k \rangle, e_i \in \mathcal{E}$ with respect to each entity's relevance to q. The relevance of entities is inferred based on a collection of unstructured and/or (semi-)structured data.

[1] Ad hoc refers to the standard form of retrieval in which the user, motivated by an ad hoc information need, initiates the search process by formulating and issuing a query

Example queries

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?

Main strategy

- Build on work on document retrieval
- Create and entity description or "profile" document is to be compiled for each entity in the catalog
  - Specifically, a fielded *entity document*
- Those entity description documents can be ranked the same way as documents

Entity retrieval

Term-based entity representations

- The key statistic is *term count*, $c(t;e)$: the number of times term t appears in the description constructed for entity e
- Other components of bag-of-words models can be derived analogously to document retrieval, e.g.,:
  - *Entity length*: $l_e = \sum_{t \in \mathcal{V}} c(t;e)$
  - *Term frequency*: $TF(t,e) = \frac{c(t;e)}{l_e}$
  - *Entity frequency*: $EF(t) = |\{e \in \mathcal{E} : c(t;e) > 0\}|$
  - *Inverse entity frequency*: $IEF(t) = \log \frac{|\mathcal{E}|}{EF(t)}$
  - where $\mathcal{E}$ is the entity catalog and $\mathcal{V}$ is the vocabulary of terms

Data collection

- Unstructured documents
- Semi-structured documents
- Structured knowledge bases

From unstructured documents

- Input: documents that are annotated with entities
  - Document-level annotations (e.g., tags)
  - Mention-level annotations (e.g,. as in Wikipedia)

> The **2004 Belgian Grand Prix** (formally the **Formula 1 Belgian Grand Prix 2004**)[2] was a Formula One motor race held on 29 August 2004, at the Circuit de Spa-Francorchamps, near the town of Spa, Belgium. It was Race 14 of 18 in the 2004 FIA Formula One World Championship. The race was contested over 44 laps and was won by Kimi Räikkönen, taking his and McLaren's only race win of the season from tenth place on the grid. Second place for Michael Schumacher won his seventh world championship, after beating third-placed Rubens Barrichello.

Figure: Excerpt from `https://en.wikipedia.org/wiki/2004_Belgian_Grand_Prix`.

- Output: (pseudo) term counts $\tilde{c}(t; e)$

From unstructured documents

- General formula, using documents as a bridge between terms and entities:

$$\tilde{c}(t; e) = \sum_{d \in \mathcal{D}} c(t, e; d)\, w(e, d)$$

- where
  - $c(t, e; d)$: number of co-occurrences between a term and an entity in a particular document
  - $w(e, d)$: strength of the *association* between the entity and the document

Using document-level annotations

- Entity description == the contents of all documents concatenated that are tagged with the given entity
- Formally:
  - Term count in the document if it mentions the entity, and are zero otherwise:

$$c(t, e; d) = \begin{cases} c(t; d), & e \in d \\ 0, & e \notin d \ . \end{cases}$$

  - Binary document-entity weights:

$$w(e, d) = \begin{cases} 1, & e \in d \\ 0, & e \notin d \ . \end{cases}$$

Using mention-level annotations

- Assume that entity occurrences have been replaced with unique identifiers that behave as regular terms
- Simplest approach: consider terms that co-occur with the entity within a fixed window size of w
- Formally:

$$c(t, e; d) = \sum_{i=1}^{l_d} {}_d(i, t) \sum_{\substack{j=1 \\ |i-j| \leq w}}^{l_d} 1_d(j, e)$$

- where
  - $1_d(i, t)$ returns 1 if the term at position i in d is t, and 0 otherwise
  - $1_d(i, e)$ returns 1 if entity e appears at position i in d, and 0 otherwise

Using mention-level annotations

- Intuition: terms closer to the mention of an entity should be given more importance than terms appearing farther away
- Formally expressed using proximity kernels

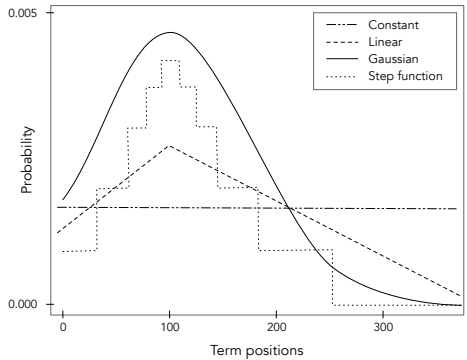


Figure: Illustration of proximity kernels from Balog (2018) [Fig. 3.3].

From semi-structured documents

- E.g., Wikipedia article, IMDB page, LikedIn profile, ...
- Field content is typically extracted using wrappers (template-based extractors)

Example

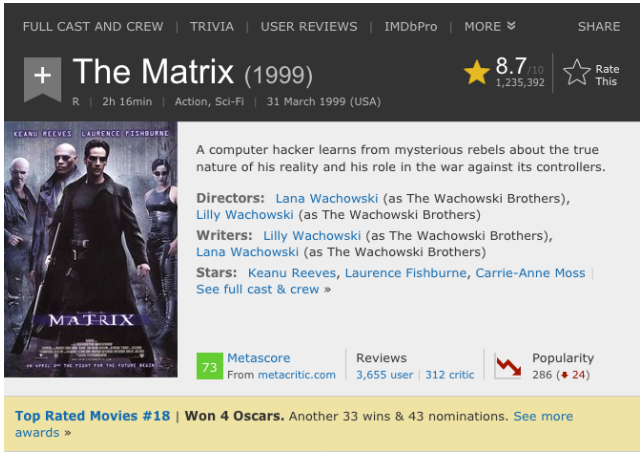


Figure: Web page of the movie The Matrix from IMDb (`http://www.imdb.com/title/tt0133093/`).

Example

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

Question

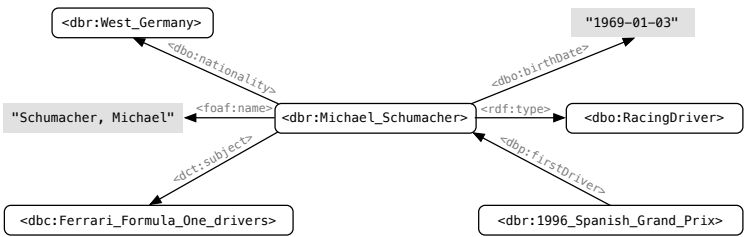
What is the role of the catch-all field?

Catch-all field

- Amasses the contents of all fields
  - Can help to quickly filter entities (e.g., in first-pass retrieval)
  - Fields are often sparse; combining field-level scores with an entity-level ("catch-all" score) often improve performance

From structured knowledge bases

- Assemble text from all SPO triples that are about a given entity
  - Note that the entity may also stand as object



Question

How to turn SPO triples into a fielded document?

Issue #1

- The number of potential fields is huge (in the 1000s)
  - The representation of an entity is sparse (each entity has only a handful of predicates)
  - Estimating field weights becomes problematic
- Solution: *predicate folding*
  - Grouping predicates together into a small set of predefined categories
  - Grouping may be based on predicate type or (manually determined) importance

Commonly used fields

- **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., top-100 most frequent DBpedia predicates
- **Catch-all** is a field that amasses all textual content related to the entity

Issue #2

- Object values are either URIs or literals
- While literals can be treated as regular text, URIs are not suitable for text-based search
  - Some URIs are "user-friendly": `http://dbpedia.org/resource/Audi_A4`
  - Others are not: `http://rdf.freebase.com/ns/m.030qmx`
- *URI resolution* is the process of finding the corresponding human-readable name/label for a URI

URI resolution

- Goal: find the name/label for a URI
- 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 an 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"
```

Example

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

Exercise

E6-2 DBpedia Trivia

Entity retrieval

Models for entity ranking

- Unstructured retrieval models
  - LM, BM25, **SDM**
- Fielded retrieval models
  - MLM, BM25F, **PRMS**, **FSDM**

Markov random field (MRF) models

- Models so far employed a bag-of-words representation of both entities and queries
  - The order of terms is ignored
- The *Markov random field* (MRF) model provides a sound theoretical framework for modeling term dependence
  - Term dependencies are represented as a Markov random field (undirected graph G)
  - The MRF ranking function is computed as a linear combination of feature functions over the set of cliques[2] in G:

$$P_\Lambda(e|q) \overset{\text{rank}}{=} \sum_{c \in \mathcal{C}_G} \lambda_c f(c)$$

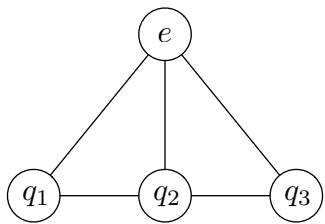
  - MRF approaches belong to the more general class of linear feature-based models

[2] A clique is a subset of vertices of an undirected graph, such that every two distinct vertices are adjacent.

Sequential Dependence Model (SDM)

- The *sequential dependence model* (SDM) is one particular instantiation of the MRF model
  - SDM assumes dependence between neighboring query terms
  - Strikes a good balance between effectiveness and efficiency

- Graph consists of an entity node e and query nodes q_i
- Two types of cliques
  - Query term and the entity (unigram matches)
  - Two query terms and the entity; two variants:
    - The query terms occur contiguously (ordered bigram match)
    - They do not (unordered bigram match)



Sequential Dependence Model (SDM)

- The SDM ranking function is given by a weighted combination of three feature functions
  - Query terms (f_T)
  - Exact match of query bigrams (f_O)
  - Unordered match of query bigrams (f_U)

$$\textit{score}(e, q) = \lambda_T \sum_{i=1}^{n} f_T(q_i, e) + \lambda_O \sum_{i=1}^{n-1} f_O(q_i, q_{i+1}, e) + \lambda_U \sum_{i=1}^{n-1} f_U(q_i, q_{i+1}, e)$$

- The query is represented as a sequence of terms $q = \langle q_1, \ldots, q_n \rangle$
- Feature weights are subject to the constraint $\lambda_T + \lambda_O + \lambda_U = 1$
  - Recommended default setting: $\lambda_T = 0.85$, $\lambda_O = 0.1$, and $\lambda_U = 0.05$

Feature functions

- Feature functions are based on language modeling estimates using Dirichlet prior smoothing
- *Unigram matches* are based on smoothed entity language models:

$$f_T(q_i, e) = \log P(q_i|\theta_e)$$

Feature functions (cont'd)

- *Ordered bigram matches*:

$$f_O(q_i, q_{i+1}, e) = \log\left(\frac{c_o(q_i, q_{i+1}, e) + \mu P_o(q_i, q_{i+1}|\mathcal{E})}{l_e + \mu}\right)$$

  - $c_o(q_i, q_{i+1}, e)$ denotes the number of times the terms q_i, q_{i+1} occur in this exact order in the description of e
  - l_e is the length of the entity's description (number of terms)
  - $\mathcal{E}$ is the entity catalog (set of all entities)
  - μ is the smoothing parameter
  - The background language model is a maximum likelihood estimate:

$$P_o(q_i, q_{i+1}|\mathcal{E}) = \frac{\sum_{e\in\mathcal{E}} c_o(q_i, q_{i+1}, e)}{\sum_{e\in\mathcal{E}} l_e} \ .$$

Feature functions (cont'd)

- *Unordered bigram matches*:

$$f_U(q_i, q_{i+1}, e) = \log\left(\frac{c_w(q_i, q_{i+1}, e) + \mu P_w(q_i, q_{i+1}|\mathcal{E})}{l_e + \mu}\right) ,$$

  - $c_w(q_i, q_{i+1}, e)$ counts the co-occurrence of terms q_i and q_{i+1} in e, within an unordered window of w term positions
  - Typically, a window size of 8 is used (corresponds roughly to sentence-level proximity)
  - l_e is the length of the entity's description (number of terms)
  - $\mathcal{E}$ is the entity catalog (set of all entities)
  - μ is the smoothing parameter
  - The background language model is a maximum likelihood estimate:

$$P_w(q_i, q_{i+1}|\mathcal{E}) = \frac{\sum_{e\in\mathcal{E}} c_w(q_i, q_{i+1}, e)}{\sum_{e\in\mathcal{E}} l_e} .$$

Example

Entity description	a b c a b c c d e f b c f e g
Query (q)	a b c
Ordered bigram matches: $c_o(q_i, q_{i+1}, e)$	
$c_o(\mathsf{a}, \mathsf{b}, e)$	2
$c_o(\mathsf{b}, \mathsf{c}, e)$	3 ⇒ a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g
Unordered bigram matches: $c_w(q_i, q_{i+1}, e)$ $(w = 5)$	
$w_o(\mathsf{a}, \mathsf{b}, e)$	4
$w_o(\mathsf{b}, \mathsf{c}, e)$	7 ⇒ a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g a b c a b c c d e f b c f e g

Question

How would you implement the SDM scoring function on top of Elasticsearch?

Exercise

E6-3 Counting Bigram Matches

Models for entity ranking

- ~~Unstructured retrieval models~~
  - ~~LM, BM25, **SDM**~~
- Fielded retrieval models $\Leftarrow$
  - MLM, BM25F, **PRMS**, **FSDM**

Mixture of Language Models (MLM)

- Idea: Build a separate language model for each field, then take a linear combination of them

$$P(t|\theta_d) = \sum_i w_i P(t|\theta_{d_i})$$

- where
  - i corresponds to the field index
  - w_i is the field weight (such that $\sum_i w_i = 1$)
  - $P(t|\theta_{d_i})$ is the field language model

Probabilistic Retrieval Model for Semistructured data (PRMS)

- Extension to MLM for dynamic field weighting
- To key ideas
  - Instead of using a fixed (static) field weight for all terms, field weights are determined dynamically on a term-by-term basis
  - Field weights can be established based on the term distributions of the respective fields
- Replace the static weight w_i with a *mapping probability* $P(f|t)$

$$P(t|\theta_d) = \sum_f P(f|t)P(t|\theta_{d_f})$$

  - Note: we now use field f instead of index i when referring to fields

Estimating the mapping probability

- By applying Bayes' theorem and using the law of total probability:

$$P(f|t) = \frac{P(t|f)P(f)}{P(t)} = \frac{P(t|f)P(f)}{\sum_{f' \in \mathcal{F}} P(t|f')P(f')}$$

- where
  - $P(f)$ is a prior that can be used to incorporate, for example, domain-specific background knowledge, or left to be uniform
  - $P(t|f)$ is conveniently estimated using the background language model of that field $P(t|C_f)$

Example

$t =$ ``Meg"		$t =$ ``Ryan"		$t =$ ``war"		$t =$ ``redemption"	
f	$P(f\|t)$	f	$P(f\|t)$	f	$P(f\|t)$	f	$P(f\|t)$
cast	0.407	cast	0.601	genre	0.927	title	0.983
team	0.381	team	0.381	title	0.070	location	0.017
title	0.187	title	0.017	location	0.002	year	0.000

Table: Example mapping probabilities computed on the IMDB collection, taken from Kim et al., 2009.

Fielded Sequential Dependence Model (FSDM)

- Idea: base the feature function estimates on term/bigram frequencies combined across multiple fields (in the spirit of MLM and BM25F)
- The *fielded sequential dependence model* (FSDM) we present here combines SDM and MLM
- *Unigram matches* are MLM-estimated probabilities:

$$f_T(q_i, e) = \log \sum_{f \in \mathcal{F}} w_f^T \, P(t|\theta_{f_e})$$

  - w_f^T are the field mapping weights (for each field)

Feature functions (cont'd)

- *Unordered bigram matches*:

$$f_O(q_i, q_{i+1}, e) = \log \sum_{f \in \mathcal{F}} w_f^O \frac{c_o(q_i, q_{i+1}, f_e) + \mu_f P_o(q_i, q_{i+1} | f_\mathcal{E})}{l_{f_e} + \mu_f}$$

- *Ordered bigram matches*:

$$f_U(q_i, q_{i+1}, e) = \log \sum_{f \in \mathcal{F}} w_f^U \frac{c_u^w(q_i, q_{i+1}, f_e) + \mu_f P_u^w(q_i, q_{i+1} | f_\mathcal{E})}{l_{f_e} + \mu_f}$$

- All background models and smoothing parameters are made field-specific
  - But the same smoothing parameter (μ_f) may be used for all types of matches
- w_f^O and w_f^U are the field mapping weights (for each field)
  - May be based on the field mapping probability estimates from PRMS

Summary

- Entities
  - Named entities vs. concepts
  - Properties of entities (unique IDs, names, types, attributes, relationships)
  - Entity-oriented search vs. semantic search
  - Entity catalog, knowledge repository, knowledge base/graph
- Data sources
  - Wikipedia article structure
  - Resource Description Framework (RDF)
  - Knowledge bases (DBpedia, Wikidate, LOD)
- The ad hoc entity retrieval task
- Constructing entity description documents
  - From unstructured, semi-structured, and structured sources
  - Working with SPO triples (predicate folding and URI resolution)
- Entity ranking
  - Using standard document retrieval models (LM, BM25, MLM, BM25F)
  - Sequential Dependence Models (SDM) and fielded variant (FSDM)
  - Probabilistic Retrieval Model for Semi-Structured Data (PRMS)

Reading

- Entity-Oriented Search (Balog)
  - Chapters 1 and 2
  - Chapter 3, until 3.3.2 (inclusive)