DAT630

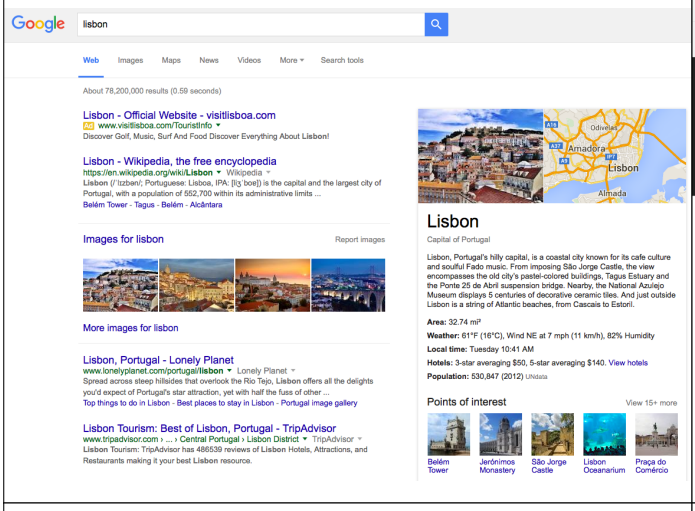
Semantic Search

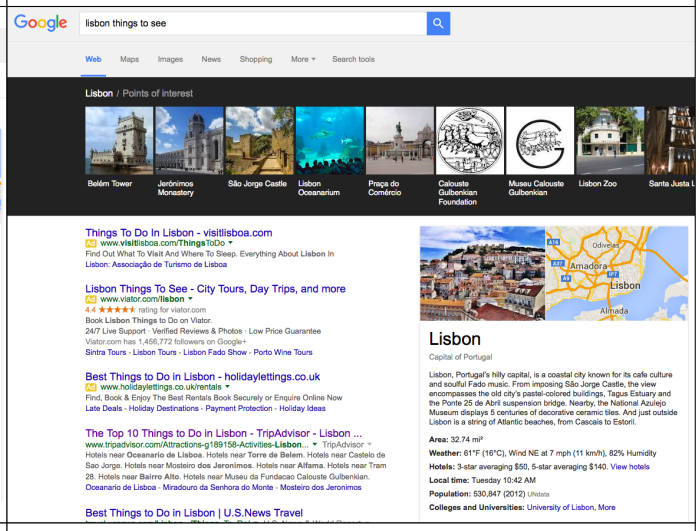
Part I, Entity Retrieval

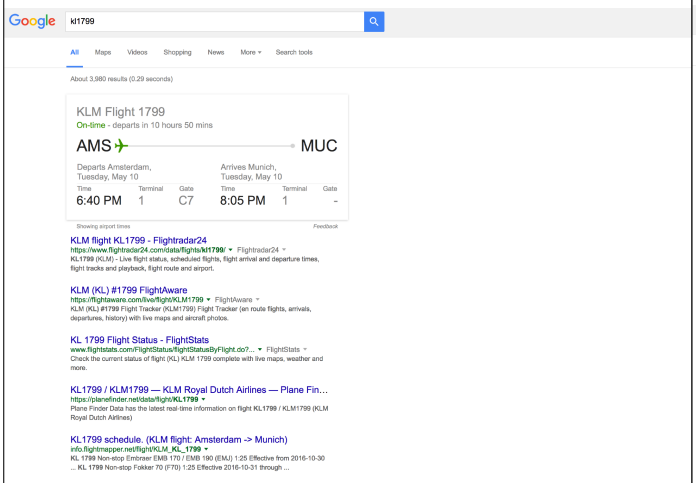
13/11/2017

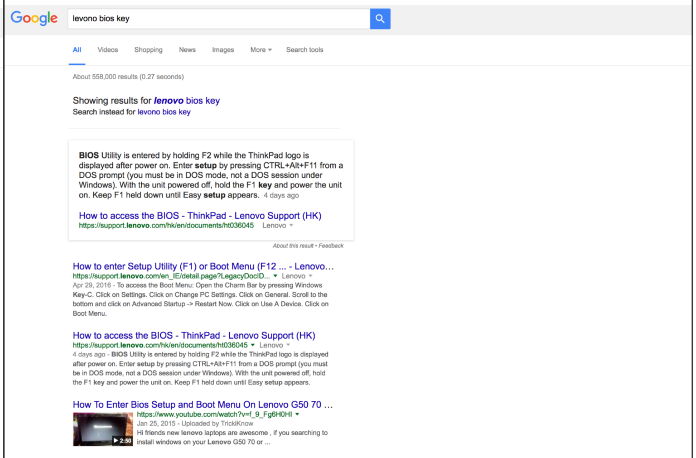
Krisztian Balog | University of Stavanger

What is semantic search?









Semantic search

- "search with meaning"
  - beyond literal matches
  - understanding what the query actually *means*

What is an entity?

people

organizations

locations

products

What is an entity?

- Uniquely identifiable *thing* or *object*
  - "A thing with a distinct and independent existence"

An entity is characterized by having…

- Unique ID
- Name(s)
- Type(s)
- Attributes (/Descriptions)
- Relationships to other entities

Entities…

- are meaningful units for organizing information
- are a key enabling component in semantic search

Outline

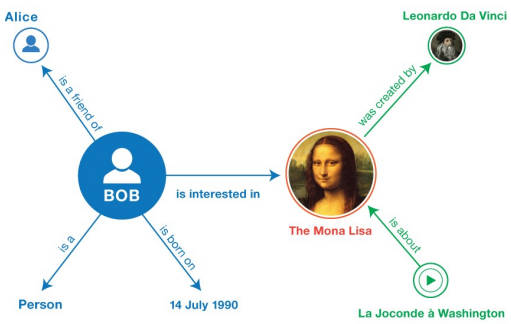
- Knowledge bases
- Two specific tasks:
  - **Entity retrieval**: given a free text query, return a ranked list of entities (instead of documents)
  - **Entity linking**: given a piece of text (e.g., document or query), recognize mentions of entities and assign to these unique identifiers from a knowledge base

Knowledge Bases

Knowledge Base

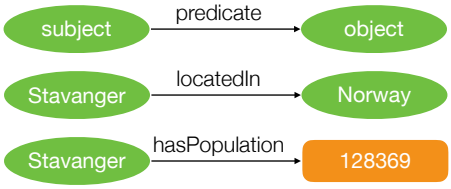
- A data repository for storing entities and their properties in a structured format
- A set of assertions about the world, describing specific entities and their relationships
- Conceptually, it forms a graph (*knowledge graph*)

Knowledge bases



RDF Data Model

- Resource Description Framework
- "Everything is a triple"
  - **Subject** (resource), **predicate** (relation), **object** (resource or literal)



Early Attempt: 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"

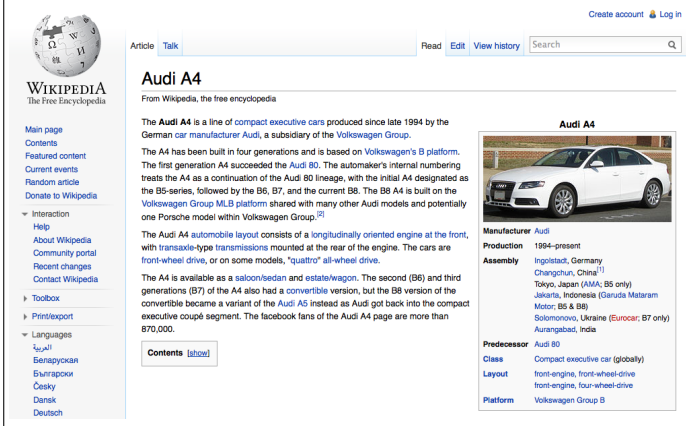
Popular (Public) Knowledge Bases

- DBpedia
- Freebase
- Wikidata
- YAGO

DBpedia

http://dbpedia.org

- Extracted from Wikipedia (mostly from infoboxes) using a set of manually constructed mapping rules
- Available in multiple languages
- Contains over 5 million entities (English)



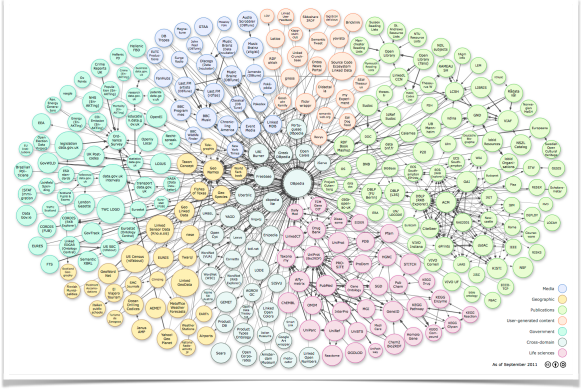

```
dbpedia:Audi_A4
```

`foaf:name`	Audi A4
`rdfs:label`	Audi A4
`rdfs:comment`	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group. The A4 has been built [...]
`dbpprop:production`	1994 2001 2005 2008
`rdf:type`	`dbpedia-owl:MeanOfTransportation` `dbpedia-owl:Automobile`
`dbpedia-owl:manufacturer`	`dbpedia:Audi`
`dbpedia-owl:class`	`dbpedia:Compact_executive_car`
`owl:sameAs`	`freebase:Audi A4`
`is dbpedia-owl:predecessor of`	`dbpedia:Audi_A5`
`is dbpprop:similar of`	`dbpedia:Cadillac_BLS`

Freebase

- Launched in 2007 by the company Metaweb
- Part of the data is imported (Wikipedia, MusicBrainz, etc.)
- Another part comes from user-submitted wiki contributions
- 1.9 billion triples about 39 million entities
- Acquired by Google in 2010
  - Used as the core of the Google Knowledge Graph
  - Shut down in 2014 (data donated to Wikidata)

Linking Open Data (LOD)

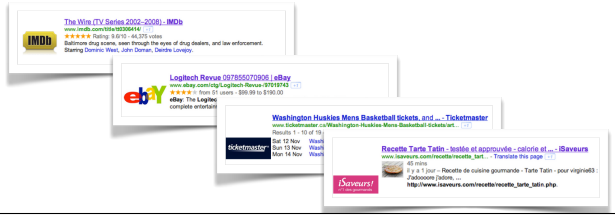


(re)Branding

- Semantic Web data
- Linking Open Data
- Web of Data

RDFa

- For embedding rich metadata within Web documents
- schema.org, sitemaps.org
  - used by Google, Bing, Yandex, Yahoo!, IPTC, etc.



Proprietary Knowledge Bases

Google Knowledge Graph

facebook Entity Graph

Microsoft Satori

> … the knowledge graph is one of Google's biggest search milestones of the last decade…
>
> —Amit Singhal, Google's director of search

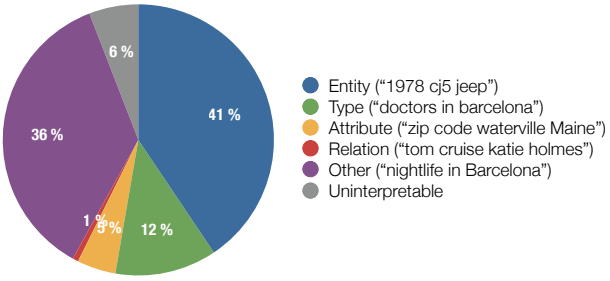
See: https://www.youtube.com/watch?v=mmQl6VGvX-c

Entity Retrieval

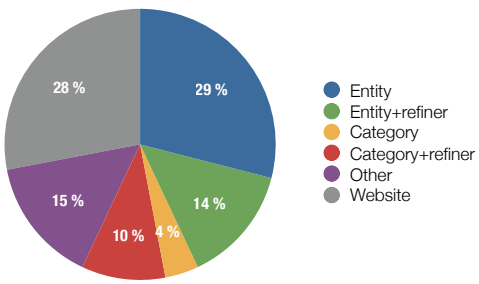
Entity retrieval

*Addressing information needs that are better answered by **returning specific objects** (entities) instead of just any type of documents.*

Distribution of web search queries [Pound et al. 2010]



Distribution of web search queries [Lin et al. 2012]



Two main scenarios

- Entity descriptions (or profile document) are readily available
  - Entity's homepage
  - Knowledge base entry
- Ready-made entity descriptions are unavailable
  - Recognize and disambiguate entities in text
  - (that is, entity linking)
  - Collect and aggregate information about a given entity from multiple documents (and even multiple data collections)

Examples of entity homepages

Ranking entities using ready-made representations

- In this scenario, ranking entities is much like ranking documents
  - unstructured
  - semi-structured

Mixture of Language Models

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

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

Field weights

$$\sum_{j=1}^{m} \mu_j = 1$$

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

Setting field weights

- Heuristically
  - Proportional to the length of text content in that field, to the field's individual performance, etc.
- Empirically (using training queries)
- Problems
  - Number of possible fields is huge
    - It is not possible to optimize their weights directly
- Entities are sparse w.r.t. different fields
  - Most entities have only a handful of predicates

Predicate folding

- **Idea**: reduce the number of fields by grouping them together
- Grouping based on
  - type
  - manually determined importance

Predicate folding

Name	`foaf:name`	Audi A4
	`rdfs:label`	Audi A4
Attributes	`rdfs:comment`	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group. The A4 has been built [...]
	`dbpprop:production`	1994 2001 2005 2008
Out-relations	`rdf:type`	`dbpedia-owl:MeanOfTransportation` `dbpedia-owl:Automobile`
	`dbpedia-owl:manufacturer`	`dbpedia:Audi`
	`dbpedia-owl:class`	`dbpedia:Compact_executive_car`
	`owl:sameAs`	`freebase:Audi_A4`
In-relations	`is dbpedia-owl:predecessor of`	`dbpedia:Audi_A5`
	`is dbpprop:similar of`	`dbpedia:Cadillac_BLS`

Setting field weights

- So far:
  - Field weights need to be set manually
  - Fields weights are the same for all query terms
- Can we estimate the field weights automatically for each query term?

Probabilistic Retrieval Model for Semistructured data

- Extension to the Mixture of Language Models
- Find which document field each query term may be associated with

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

Mapping probability
Estimated for each query term

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

Estimating the mapping probability

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

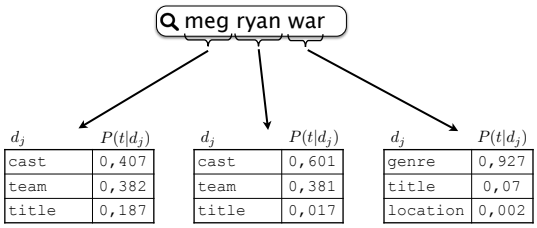
Term likelihood
Probability of a query term occurring in a given field type

Prior field probability
Probability of mapping the query term to this field before observing collection statistics

$$P(d_j|t) = \frac{\overbrace{P(t|d_j)}\overbrace{P(d_j)}}{\underbrace{P(t)}}$$

$$\sum_{d_k} P(t|d_k)P(d_k)$$

Example



d_j	$P(t\|d_j)$
cast	0,407
team	0,382
title	0,187

d_j	$P(t\|d_j)$
cast	0,601
team	0,381
title	0,017

d_j	$P(t\|d_j)$
genre	0,927
title	0,07
location	0,002

DAT630

Semantic Search

Part II, Entity Linking

13/11/2017

Krisztian Balog | University of Stavanger

From Named Entity Recognition to Entity Linking

Named entity recognition (NER)

- Also known as *entity identification*, *entity extraction*, and *entity chunking*
- Task: identifying named entities in text and labeling them with one of the possible entity types
  - Person (PER), organization (ORG), location (LOC), miscellaneous (MISC). Sometimes also temporal expressions (TIMEX) and certain types of numerical expressions (NUMEX)

```
<LOC>Silicon Valley</LOC> venture capitalist <PER>Michael
Moritz</PER> said that today's billion-dollar "unicorn" startups
can learn from <ORG>Apple</ORG> founder <PER>Steve Jobs</PER>
```

Named entity disambiguation

- Also called *named entity normalization* and *named entity resolution*
- Task: assign ambiguous entity names to canonical entities from some catalog
  - It is usually assumed that entities have already been recognized in the input text (i.e., it has been processed by a NER system)

Wikification

- Named entity disambiguation using Wikipedia as the catalog of entities
- Also annotating *concepts*, not only entities

World War II

From Wikipedia, the free encyclopedia

World War II (often abbreviated to **WWII** or **WW2**), also known as the **Second World War**, was a global war that lasted from 1939 to 1945, although related conflicts began earlier. It involved the vast majority of the world's nations—including all of the great powers—eventually forming two opposing military alliances: the Allies and the Axis. It was the most widespread war in history, and directly involved more than 100 million people from over 30 countries. In a state of "total war", the major participants threw their entire economic, industrial, and scientific capabilities behind the war effort, erasing the distinction between civilian and military resources. Marked by mass deaths of civilians, including the Holocaust (in which approximately 11 million people were killed)[1][2] and the strategic bombing of industrial and population centres (in which approximately one million were killed, and which included the

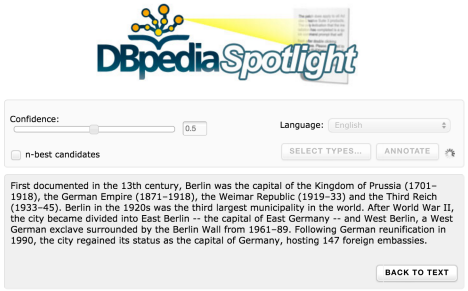
Entity linking

- Task: recognizing entity mentions in text and linking them to the corresponding entries in a knowledge base (KB)
  - Limited to recognizing entities for which a target entry exists in the reference KB; each KB entry is a candidate
  - It is assumed that the document provides sufficient context for disambiguating entities
- Knowledge base (working definition):
  - A catalog of entities, each with one or more names (surface forms), links to other entities, and, optionally, a textual description
  - Wikipedia, DBpedia, Freebase, YAGO, etc.

Overview of entity annotation tasks

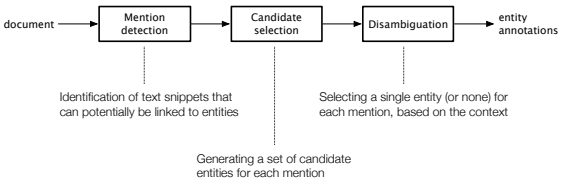
Task	Recognition	Assignment
Named entity recognition	entities	entity type
Named entity disambiguation	entities	entity ID / NIL
Wikification	entities and concepts	entity ID / NIL
Entity linking	entities	entity ID

Entity linking in action

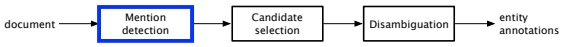


Entity linking in action

Anatomy of an entity linking system



Mention detection



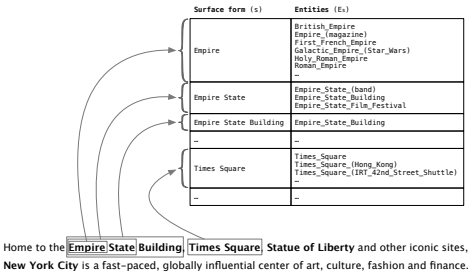
Mention detection

- Goal: Detect all “linkable” phrases
- Challenges:
  - Recall oriented
    - Do not miss any entity that should be linked
  - Find entity name variants
    - E.g. “jlo” is name variant of [Jennifer Lopez]
  - Filter out inappropriate ones
    - E.g. “new york” matches >2k different entities

Common approach

1. Build a dictionary of entity surface forms
   - Entities with all names variants
2. Check all document n-grams against the dictionary
   - The value of n is set typically between 6 and 8
3. Filter out undesired entities
   - Can be done here or later in the pipeline

Example



Surface form dictionary construction from Wikipedia

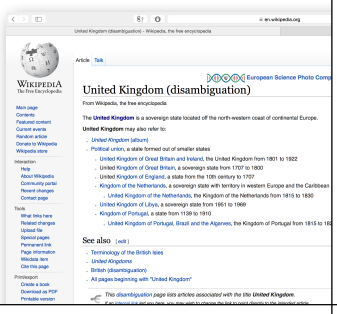
- **Page title**
  - Canonical (most common) name of the entity

Surface form dictionary construction from Wikipedia

- Page title
- **Redirect pages**
  - Alternative names that are frequently used to refer to an entity

Surface form dictionary construction from Wikipedia

- Page title
- Redirect pages
- **Disambiguation pages**
  - List of entities that share the same name



Surface form dictionary construction from Wikipedia

- Page title
- Redirect pages
- Disambiguation pages
- **Anchor texts**
  - of links pointing to the entity's Wikipedia page

Surface form dictionary construction from Wikipedia

- Page title
- Redirect pages
- Disambiguation pages
- Anchor texts
- **Bold texts from first paragraph**
  - generally denote other name variants of the entity

Surface form dictionary construction from other sources

- Anchor texts from external web pages pointing to Wikipedia articles
- Problem of *synonym discovery*
  - Expanding acronyms
  - Leveraging search results or query-click logs from a web search engine
  - ...

Filtering mentions

- Filter our mentions that are unlikely to be linked to any entity
- **Keyphraseness**:

$$P(\text{keyphrase}|m) = \frac{|D_{link}(m)|}{|D(m)|}$$

$|D_{link}(m)|$ → number of Wikipedia articles where m appears as a link

$|D(m)|$ → number of Wikipedia articles that contain m

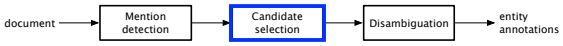
- **Link probability**:

$$P(\text{link}|m) = \frac{\text{link}(m)}{\text{freq}(m)}$$

$\text{link}(m)$ → the number of times mention m appears as a link

$\text{freq}(m)$ → total number of times mention m occurs in Wikipedia (as a link or not)

Candidate selection



Overlapping entity mentions

- Dealing with them in this phase
  - E.g., by dropping a mention if it is subsumed by another mention
- Keeping them and postponing the decision to a later stage (candidate selection or disambiguation)

Candidate selection

- Goal: Narrow down the space of disambiguation possibilities
- Balances between precision and recall (effectiveness vs. efficiency)
- Often approached as a ranking problem
  - Keeping only candidates above a score/rank threshold for downstream processing

Commonness

- Perform the ranking of candidate entities based on their overall popularity, i.e., "most common sense"

$$P(e|m) = \frac{n(m,e)}{\sum_{e'} n(m,e')}$$

$n(m,e)$ → the number of times entity e is the link destination of mention m

$\sum_{e'} n(m,e')$ → total number of times mention m appears as a link

Example

Entity (e)	Commonness (P(e|m))
Times_Square	0.940
Times_Square_(film)	0.017
Times_Square_(Hong_Kong)	0.011
Times_Square_(IRT_42nd_Street_Shuttle)	0.006
…	…

Home to the **Empire State Building**, **Times Square**, **Statue of Liberty** and other iconic sites, **New York City** is a fast-paced, globally influential center of art, culture, fashion and finance.

Commonness

- Can be pre-computed and stored in the entity surface form dictionary
- Follows a power law with a long tail of extremely unlikely senses; entities at the tail end of the distribution can be safely discarded
  - E.g., 0.001 is a sensible threshold

Example

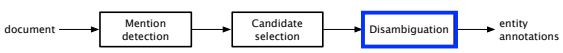
Entity	Commonness
FIFA_World_Cup	0.2358
FIS_Apline_Ski_World_Cup	0.0682
2009_FINA_Swimming_World_Cup	0.0633
World_Cup_(men's_golf)	0.0622
…	

Bulgaria's best **World Cup** performance was in the **1994 World Cup** where they beat **Germany**, to reach the semi-finals, losing to Italy, and finishing in fourth ...

Entity	Commonness
1998_FIFA_World_Cup	0.9556
1998_IAAF_World_Cup	0.0296
1998_Alpine_Skiing_World_Cup	0.0059
…	

Entity	Commonness
Germany	0.9417
Germany_national_football_team	0.0139
Nazi_Germany	0.0081
German_Empire	0.0065
…	

Disambiguation



Disambiguation

- Baseline approach: most common sense
- Consider additional types of evidence
  - **Prior importance** of entities and mentions
  - **Contextual similarity** between the text surrounding the mention and the candidate entity
  - **Coherence** among all entity linking decisions in the document
- Combine these signals
  - Using supervised learning or graph-based approaches
- Optionally perform pruning
  - Reject low confidence or semantically meaningless annotations

Prior importance features

- **Context-independent** features
  - Neither the text nor other mentions in the document are taken into account
- Keyphraseness
- Link probability
- Commonness

Prior importance features

- **Link prior**
  - Popularity of the entity measured in terms of incoming links

$$P_{\text{link}}(e) = \frac{\text{link}(e)}{\sum_{e'} \text{link}(e')}$$

- **Page views**
  - Popularity of the entity measured in terms traffic volume

$$P_{\text{pageviews}}(e) = \frac{\text{pageviews}(e)}{\sum_{e'} \text{pageviews}(e')}$$

Contextual features

- Compare the surrounding **context** of a mention with the (textual) representation of the given candidate entity
- Context of a mention
  - Window of text (sentence, paragraph) around the mention
  - Entire document
- Entity's representation
  - Wikipedia entity page, first description paragraph, terms with highest TF-IDF score, etc.
  - Entity's description in the knowledge base

Contextual similarity

- Commonly: bag-of-words representation
- **Cosine similarity**

$$\text{sim}_{cos}(m, e) = \frac{\vec{d}_m \cdot \vec{d}_e}{\| \vec{d}_m \| \| \vec{d}_e \|}$$

- Many other options for measuring similarity
  - Dot product, KL divergence, Jaccard similarity
- Representation does not have to be limited to bag-of-words
  - Concept vectors (named entities, Wikipedia categories, anchor text, keyphrases, etc.)

Entity-relatedness features

- It can reasonably be assumed that a document focuses on one or at most a few topics
- Therefore, entities mentioned in a document should be topically related to each other
- Capturing **topical coherence** by developing some measure of **relatedness** between (linked) entities
  - Defined for pairs of entities

Wikipedia Link-based Measure (WSM)

- Often referred to simply as *relatedness*
- A close relationship is assumed between two entities if there is a large overlap between the entities linking to them

$$\text{WLM}(e, e') = 1 - \frac{\log(\max(|L_e|, |L_{e'}|)) - \log(|L_e \cap L_{e'}|)}{\log(|E|) - \log(\min(|L_e|, |L_{e'}|))}$$

$|E|$: total number of entities; $|L_e|$, $|L_{e'}|$: set of entities that link to e

Wikipedia Link-based Measure (WLM)

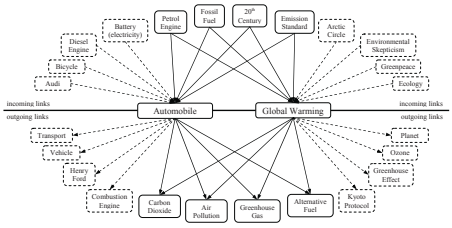


Image taken from **Milne and Witten (2008a). An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links.** In *AAAI WikiAI Workshop*.

Asymmetric relatedness features

- A relatedness function does not have to be symmetric
  - E.g., the relatedness of the UNITED STATES given NEIL ARMSTRONG is intuitively larger than the relatedness of NEIL ARMSTRONG given the UNITED STATES
- **Conditional probability**

$$P(e'|e) = \frac{|L_{e'} \cap L_e|}{|L_e|}$$

Entity-relatedness features

- Numerous ways to define relatedness
  - Consider not only incoming, but also outgoing links or the union of incoming and outgoing links
  - Jaccard similarity, Pointwise Mutual Information (PMI), or the Chi-square statistic, etc.
- Having a single relatedness function is preferred, to keep the disambiguation process simple
- Various relatedness measures can effectively be combined into a single score using a machine learning approach [Ceccarelli et al., 2013]

Overview of features

Category	Feature	Description
Prior importance (context-independent)		
	$P(\text{keyphrase}\|m)$	Keyphraseness (likelihood of m being linked)
	$P(\text{link}\|m)$	Link probability (likelihood of m being linked)
	$P(e\|m)$	Commonness (the probability of e being the link target of m)
	$P_{\text{link}}(e)$	Fraction of links in the KB pointing to e
	$P_{\text{pageviews}}(e)$	Fraction of (Wikipedia) page views e receives
Contextual		
	$\text{sim}_f(m, e)$	Similarity between the the context of a mention d_m and the entity's description d_e; the similarity function f can be cosine, Jaccard, dot product, KL divergence, etc.
Entity-relatedness		
	$\text{WLM}(e, e')$	Milne and Witten's Wikipedia Link-based Measure, a.k.a. relatedness
	$\text{PMI}(e, e')$	Pointwise Mutual Information
	$\text{Jaccard}(e, e')$	Jaccard similarity
	$\chi^2(e, e')$	χ^2 statistic
	$P(e'\|e)$	Conditional probability

Disambiguation approaches

- Consider **local compatibility** (including prior evidence) and **coherence** with the other entity linking decisions
- Task:

$$\Gamma : M_d \rightarrow E \bigcup \{\emptyset\}$$

- Objective function:

$$\Gamma^* = \arg\max_{\Gamma} \Big(\sum_{(m,e)\in\Gamma} \phi(m,e) + \psi(\Gamma) \Big)$$

local compatibility between the mention and the assigned entity

coherence function for all entity annotations in the document

This optimization problem is NP-hard!
Need to resort to approximation algorithms and heuristics

Disambiguation strategies

- **Individually**, one-mention-at-a-time
  - Rank candidates for each mention, take the top ranked one (or NIL)
  - Interdependence between entity linking decisions may be incorporated in a pairwise fashion

$$\Gamma(m) = \arg\max_{e\in E_m} \text{score}(m,e)$$

- **Collectively**, all mentions in the document jointly

Disambiguation approaches

Approach	Context	Entity interdependence
Most common sense	none	none
Individual local disambiguation	text	none
Individual global disambiguation	text & entities	pairwise
Collective disambiguation	text & entities	collective

Individual local disambiguation

- Early entity linking approaches
- Local compatibility score can be written as a linear combination of features

$$\phi(e,m) = \sum_i \lambda_i f_i(e,m)$$

Can be both context-independent and context-dependent features

- Learn the "optimal" combination of features from training data using machine learning

Individual global disambiguation

- Consider what other entities are mentioned in the document
- True global optimization would be NP-hard
- Good approximation can be computed efficiently by considering pairwise interdependencies for each mention independently
  - Pairwise entity relatedness scores need to be aggregated into a single number (how coherent the given candidate entity is with the rest of the entities in the document)

TAGME

[Ferragina & Scaiella, 2010]

- Combine the two most important features (*commonness* and *relatedness*) using a voting scheme
- The score of a candidate entity for a particular mention:

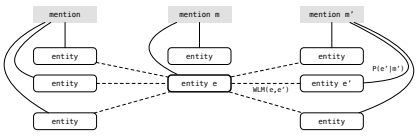
$$\text{score}(m,e) = \sum_{m'\in M_d\setminus\{m\}} \text{vote}(m',e)$$

- The vote function estimates the agreement between e and all candidate entities of all other mentions in the document

TAGME (voting mechanism)

- Average relatedness between each possible disambiguation, weighted by its commonness score

$$\text{vote}(m',e) = \frac{\sum_{e'\in E_{m'}} \text{WLM}(e,e')P(e'|m')}{|E_{m'}|}$$



TAGME (final score)

- Final decision uses a simple but robust heuristic
  - The top entities with the highest score are considered for a given mention and the one with the highest commonness score is selected

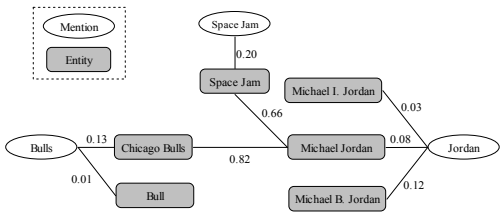
$$\Gamma(m) = \arg\max_{e\in E_m} \{P(e|m) : e \in \text{top}_\epsilon[\text{score}(m,e)]\}$$

$$\text{score}(m,e) = \sum_{m'\in M_d\setminus\{m\}} \text{vote}(m',e)$$

Collective disambiguation

- Graph-based representation
- **Mention-entity edges** capture the local compatibility between the mention and the entity
  - Measured using a combination of context-independent and context-dependent features
- **Entity-entity edges** represent the semantic relatedness between a pair of entities
  - Common choice is *relatedness* (WLM)
- Use these relations jointly to identify a single referent entity (or none) for each of the mentions

Example



AIDA

[Hoffart et al., 2011]

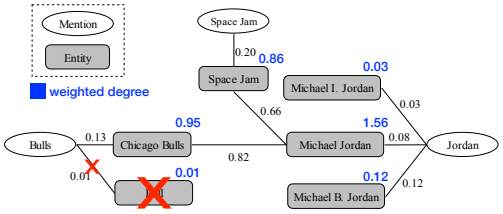
- Problem formulation: find a dense subgraph that contains all mention nodes and exactly one mention-entity edge for each mention
- Greedy algorithm iteratively removes edges

Algorithm

- Start with the full graph
- Iteratively remove the entity node with the lowest *weighted degree* (along with all its incident edges), provided that each mention node remains connected to at least one entity
  - Weighted degree of an entity node is the sum of the weights of its incident edges
- The graph with the highest *density* is kept as the solution
  - The density of the graph is measured as the minimum weighted degree among its entity nodes

Example iteration #1

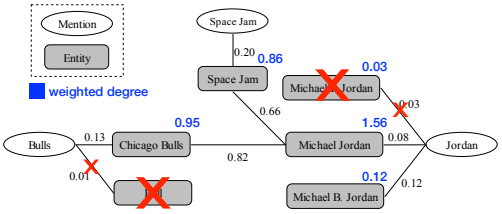
Which entity should be removed?



What is the *density* of the graph? **0.03**

Example iteration #2

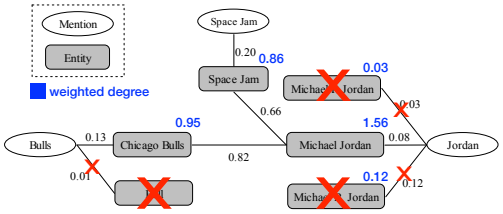
Which entity should be removed?



What is the *density* of the graph? **0.12**

Example iteration #3

Which entity should be removed?



What is the *density* of the graph? **0.86**

Pre- and post-processing

- Pre-processing phase: remove entities that are "too distant" from the mention nodes
- At the end of the iterations, the solution graph may still contain mentions that are connected to more than one entity; deal with this in post-processing
  - If the graph is sufficiently small, it is feasible to exhaustively consider all possible mention-entity pairs
  - Otherwise, a faster local (hill-climbing) search algorithm may be used

Pruning

- Discarding meaningless or low-confidence annotations produced by the disambiguation phase
- Simplest solution: use a confidence threshold
- More advanced solutions
  - Machine learned classifier to retain only entities that are ``relevant enough'' (human editor would annotate them)
  - Optimization problem: decide, for each mention, whether switching the top ranked disambiguation to NIL would improve the objective function

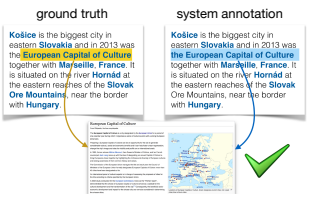
Evaluation

Evaluation (end-to-end)

- Comparing the system-generated annotations against a human-annotated gold standard
- Evaluation criteria
  - **Perfect match**: both the linked entity and the mention offsets must match
  - **Relaxed match**: the linked entity must match, it is sufficient if the mention overlaps with the gold standard

Evaluation with relaxed match

Example #1



Example #2

Evaluation metrics

- Set-based metrics:
  - **Precision**: fraction of correctly linked entities that have been annotated by the system
  - **Recall**: fraction of correctly linked entities that should be annotated
  - **F-measure**: harmonic mean of precision and recall
- Metrics are computed over a collection of documents
  - Micro-averaged: aggregated across mentions
  - Macro-averaged: aggregated across documents

Evaluation metrics

annotations generated by the entity linking system ↑ | ground truth annotations ↑

- Micro-averaged

$$\mathrm{P}_{mic} = \frac{|A_D \cap \hat{A}_D|}{|A_D|} \qquad \mathrm{R}_{mic} = \frac{|A_D \cap \hat{A}_D|}{|\hat{A}_D|}$$

- Macro-averaged

$$\mathrm{P}_{mac} = \sum_{d \in D} \frac{|A_d \cap \hat{A}_d|}{|A_d|} / |D| \qquad \mathrm{R}_{mac} = \sum_{d \in D} \frac{|A_d \cap \hat{A}_d|}{|\hat{A}_d|} / |D|$$

- F1 score

$$\mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

Component-based evaluation

- The pipeline architecture makes the evaluation of entity linking systems especially challenging
  - The main focus is on the disambiguation component, but its performance is largely influenced by the preceding steps
- Fair comparison between two approaches can only be made if they share all other elements of the pipeline

Resources

A Cross-Lingual Dictionary for English Wikipedia Concepts

- Collecting strings (mentions) that link to Wikipedia articles on the Web
  - https://research.googleblog.com/2012/05/from-words-to-concepts-and-back.html

string (mention) ↑ | Wikipedia article ↑ | number of times the mention is linked to that article ↑

text=football	*url*	*count*
1. Association football		44,984
2. American football		23,373
⋮		

Freebase Annotations of the ClueWeb Corpora

- ClueWeb annotated with Freebase entities (by Google)
  - http://lemurproject.org/clueweb09/FACC1/
  - http://lemurproject.org/clueweb12/FACC1/

name of the document that was annotated ↑

clueweb09-en0000-00-04720.html

PDF	21089	21092	0.99763662	6.6723776e-05	/m/0600q
FDA	21303	21306	0.9998256	0.00057182228	/m/032mx
Food and Drug Administration	21312	21340	0.9998256	0.00057182228	/m/032mx

entity mention ↓ | beginning and end byte offsets ↓ | confidence given both the mention and the context ↓ | confidence given just the context (ignoring the mention) ↓ | entity ID in Freebase ↓

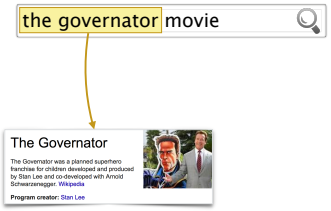
Entity linking in queries

Entity linking in queries

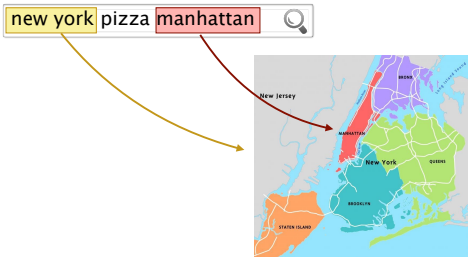
- Challenges
  - search queries are short
  - limited context
  - lack of proper grammar, spelling
  - multiple interpretations
  - needs to be fast

Example

Example



Example

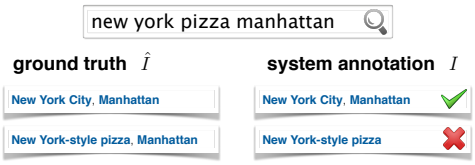


Example

ERD'14 challenge

- **Task**: finding query interpretations
- **Input**: keyword query
- **Output**: sets of sets of entities
- **Reference KB**: Freebase
- Annotations are to be performed by a web service within a given time limit

Evaluation



$$P = \frac{|I \bigcap \hat{I}|}{|I|} \qquad R = \frac{|I \bigcap \hat{I}|}{|\hat{I}|} \qquad F = \frac{2 \cdot P \cdot R}{P + R}$$

ERD'14 results

Rank	Team	F1	latency
1	SMAPH Team	0.7076	0.49
2	NTUNLP	0.6797	1.04
3	Seznam Research	0.6693	3.91

Single interpretation is returned

http://web-ngram.research.microsoft.com/erd2014/LeaderBoard.aspx