Chapter 16: Entity Search and Question Answering

Things, not Strings! -- Amit Singhal

It don't mean a thing if it ain't got that string!

-- Duke Ellington (modified)

Bing, not Thing! -- anonymous MS engineer

Search is King! -- Jürgen Geuter aka. tante

Outline

16.1 Entity Search and Ranking

16.2 Entity Linking (aka. NERD)

16.3 Natural Language Question Answering

Goal: Semantic Search

Answer „knowledge queries“
(by researchers, journalists, market & media analysts, etc.):

- **Stones? Stones songs?**
- **Dylan cover songs?**
- **African singers who covered Dylan songs?**
- **Politicians who are also scientists?**
- **European composers who have won film music awards?**
- **Relationships between**
  Niels Bohr, Enrico Fermi, Richard Feynman, Edward Teller?
  Max Planck, Angela Merkel, José Carreras, Dalai Lama?
- **Enzymes that inhibit HIV?**
  Influenza drugs for teens with high blood pressure?
  German philosophers influenced by William of Ockham?
  …..

16.1 Entity Search

Input or output of search is entities (people, places, products, etc.) or even entity-relationship structures

→ more precise queries, more precise and concise answers

	text output (docs, passages)	struct. output (entities, facts)
text input (keywords)	*Standard IR*	***Entity Search*** ***Keywords in Graphs (16.1.2)***
struct. input (entities, SPO patterns)	***Entity Search (16.1.1)***	***Semantic Web Querying (16.1.3)***

16.1.1 Entity Search with Documents as Answers

Input: one or more entities of interest
and optionally: keywords, phrases
Output: documents that contain all (or most) of
the input entities and the keywords/phrases

Typical pipeline:

1 Info Extraction: discover and mark up entities in docs

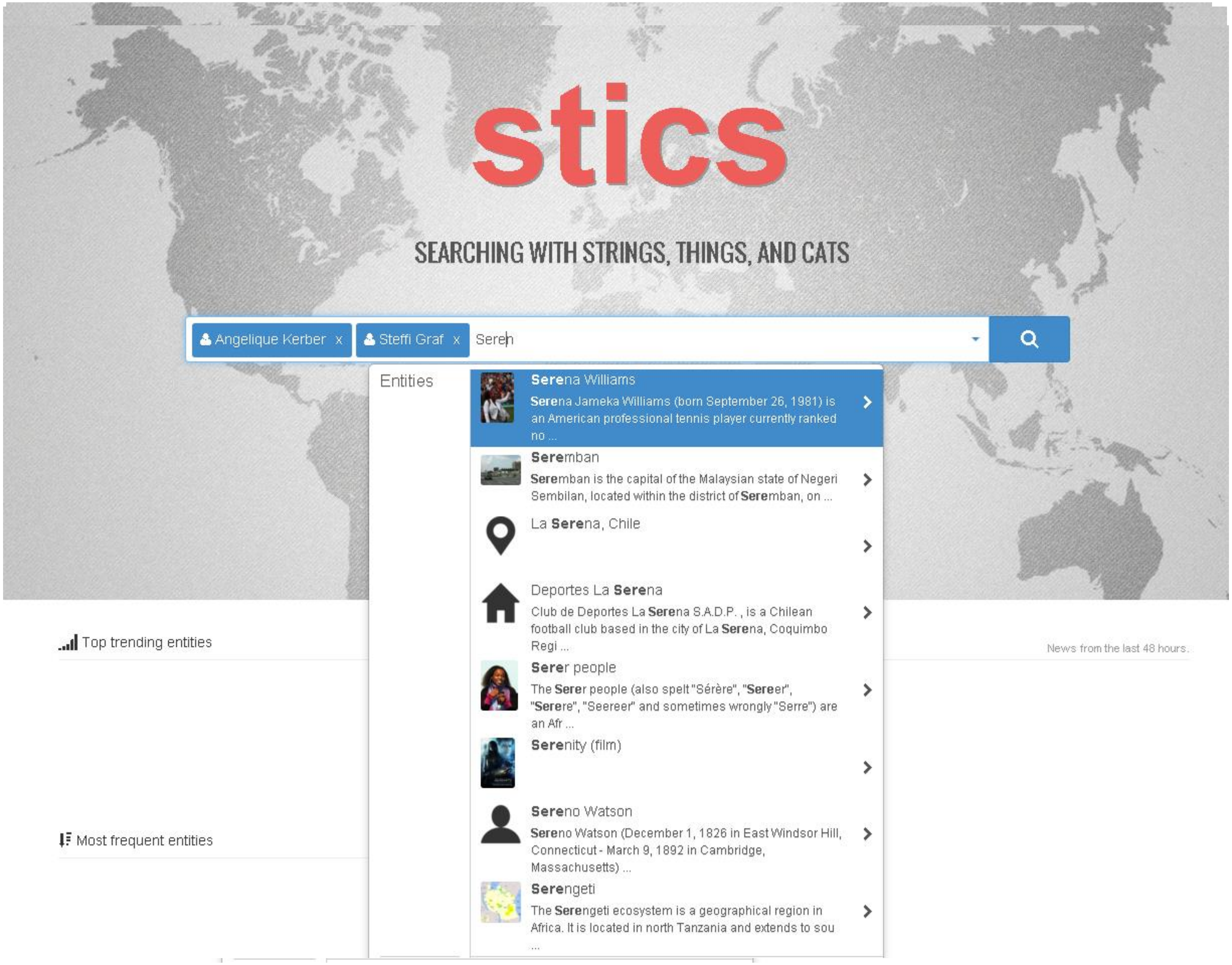
2 Indexing: build inverted list for each entity

3 Query Understanding: infer entities of interest from user input

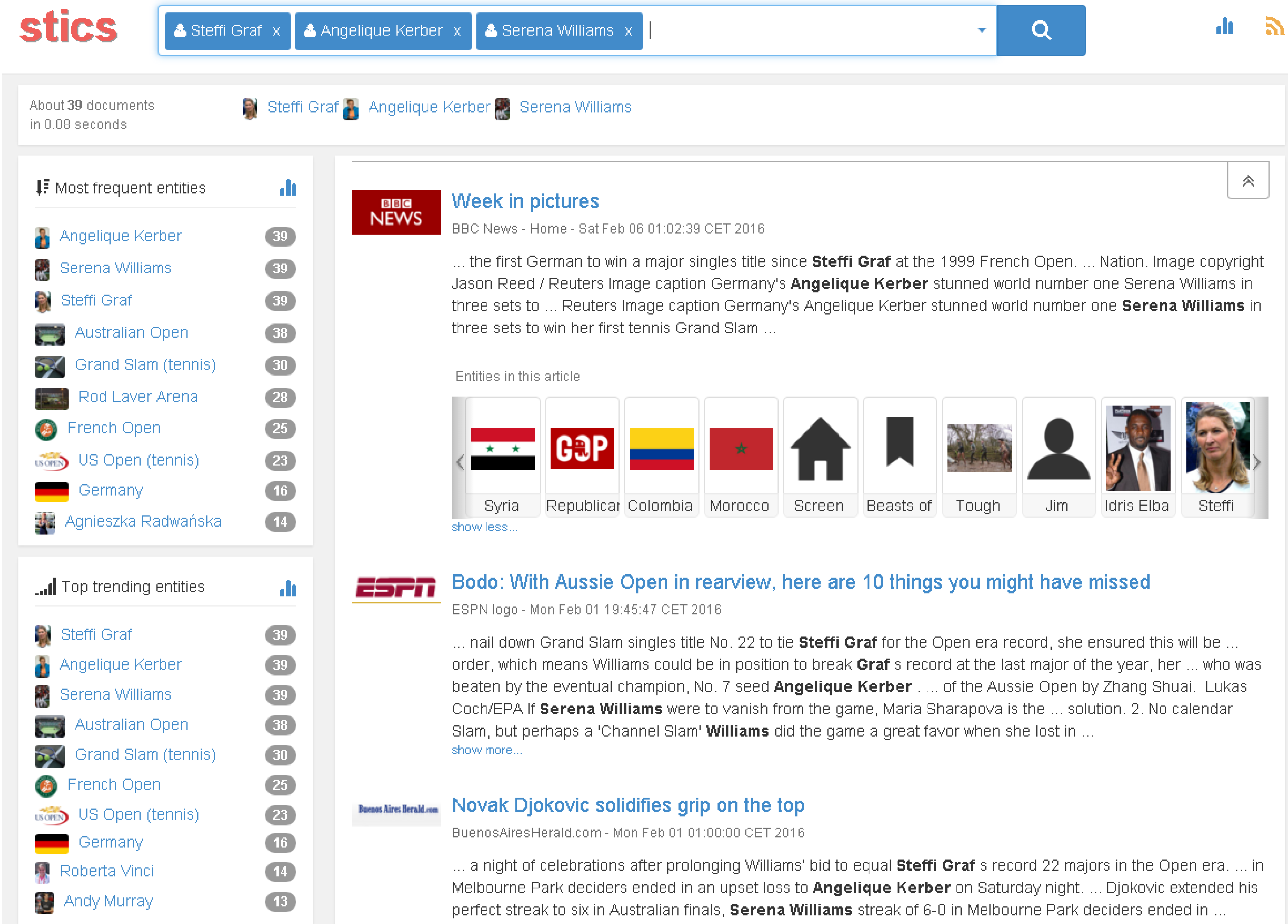
4 Query Processing: process inverted lists for entities and keywords

5 Answer Ranking: scores by per-entity LM or PR/HITS or …

Entity Search Example



Entity Search Example



Entity Search Example

Entity Search: Query Understanding

User types names → system needs to map them entities (in real-time)

Task:

given an input prefix $\boldsymbol{e_1 \ldots e_k\ x}$ with entities e_i and string x,
compute short list of auto-completion suggestions for entity e_{k+1}

Determine **candidates e** for e_{k+1} by partial matching (with indexes) against dictionary of entity alias names

Estimate for each candidate e (using precomputed statistics):

- **similarity (x, e)** by string matching (e.g. n-grams)
- **popularity (e)** by occurrence frequency in corpus (or KG)
- **relatedness (e_i, e)** for i=1..k by co-occurrence frequency

Rank and shortlist candidates e for e_{k+1} by

α similarity (x,e) + β popularity(e) + γ $\Sigma_{i=1..k}$ relatedness(e_i,e)

Entity Search: Answer Ranking

[Nie et al.: WWW'07, Kasneci et al.; ICDE'08, Balog et al. 2012]

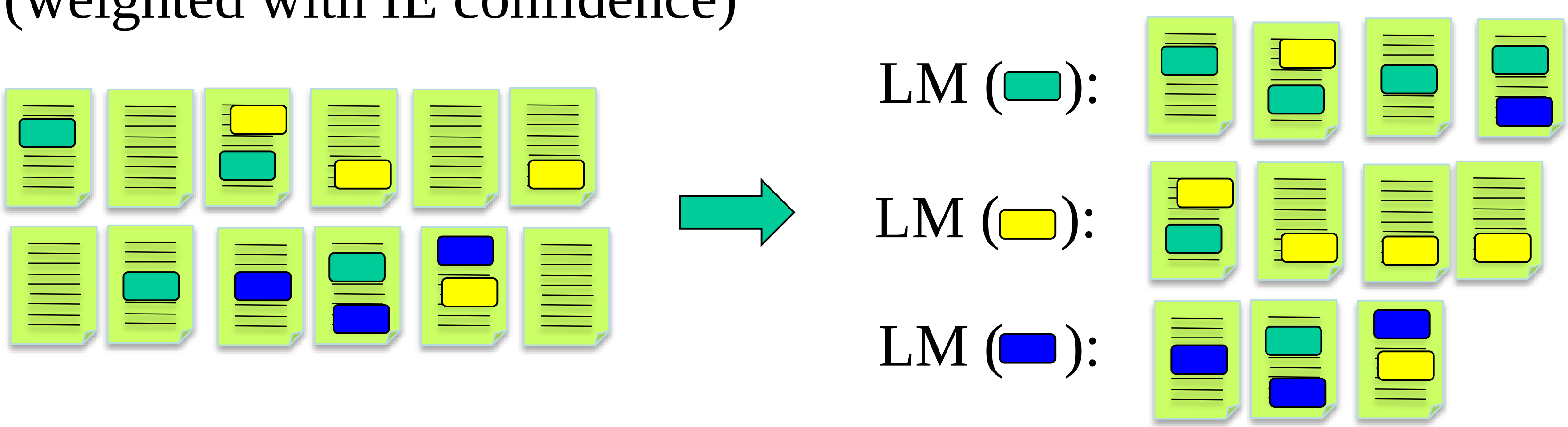
Construct language models for queries q and answers a

$$score(a,q) = \lambda P[q \mid a] + (1-\lambda) P[q] \qquad \sim KL(LM(q) \mid LM(a))$$

with smoothing

q is entity, a is doc → build LM(q): distr. on terms, by

- use IE methods to mark entities in text corpus
- associate entity with terms in docs (or doc windows) where it occurs (weighted with IE confidence)



q is keywords, a is entity → analogous

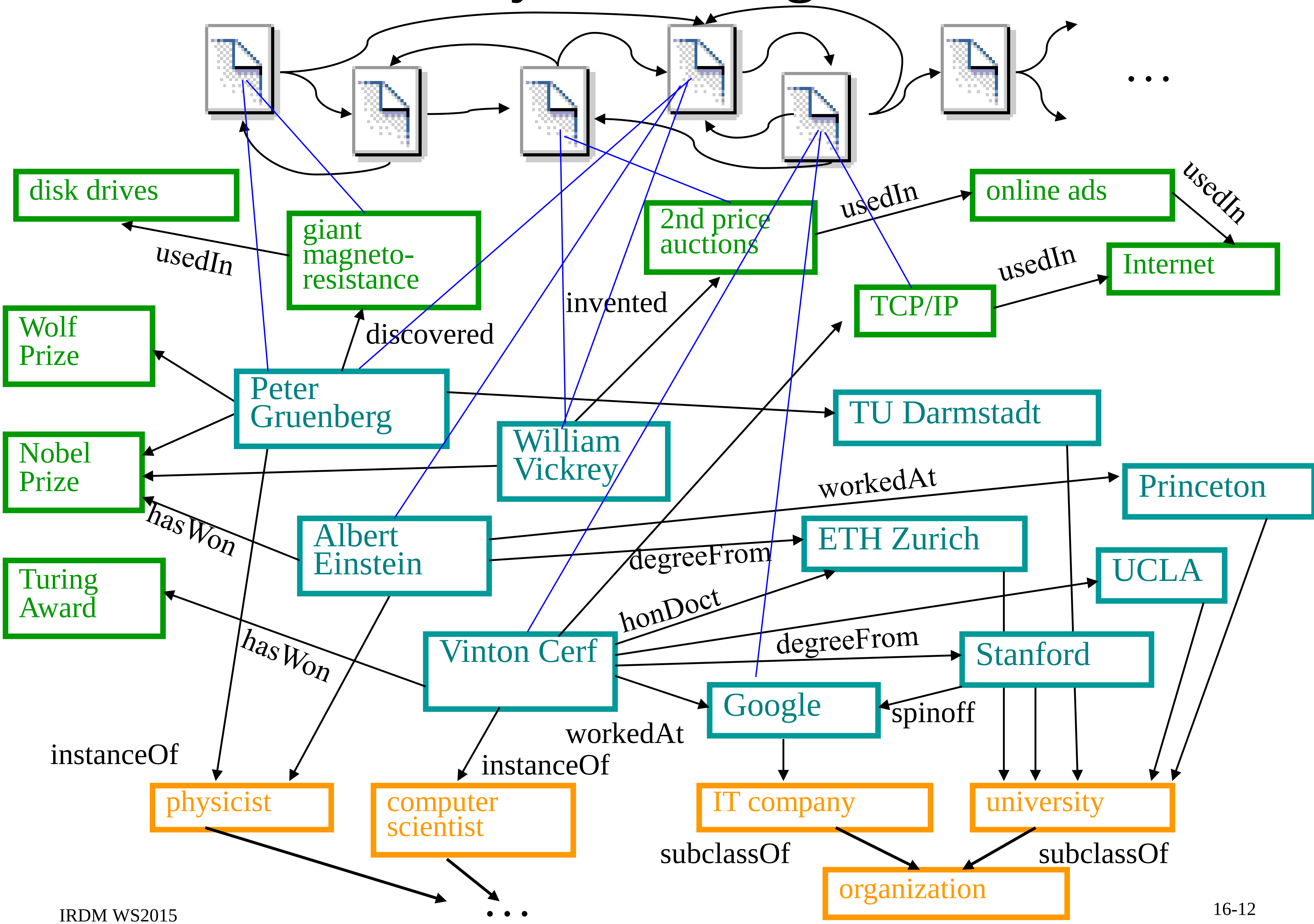
Entity Search: Answer Ranking by Link Analysis

[A. Balmin et al. 2004, Nie et al. 2005, Chakrabarti 2007, J. Stoyanovich 2007]

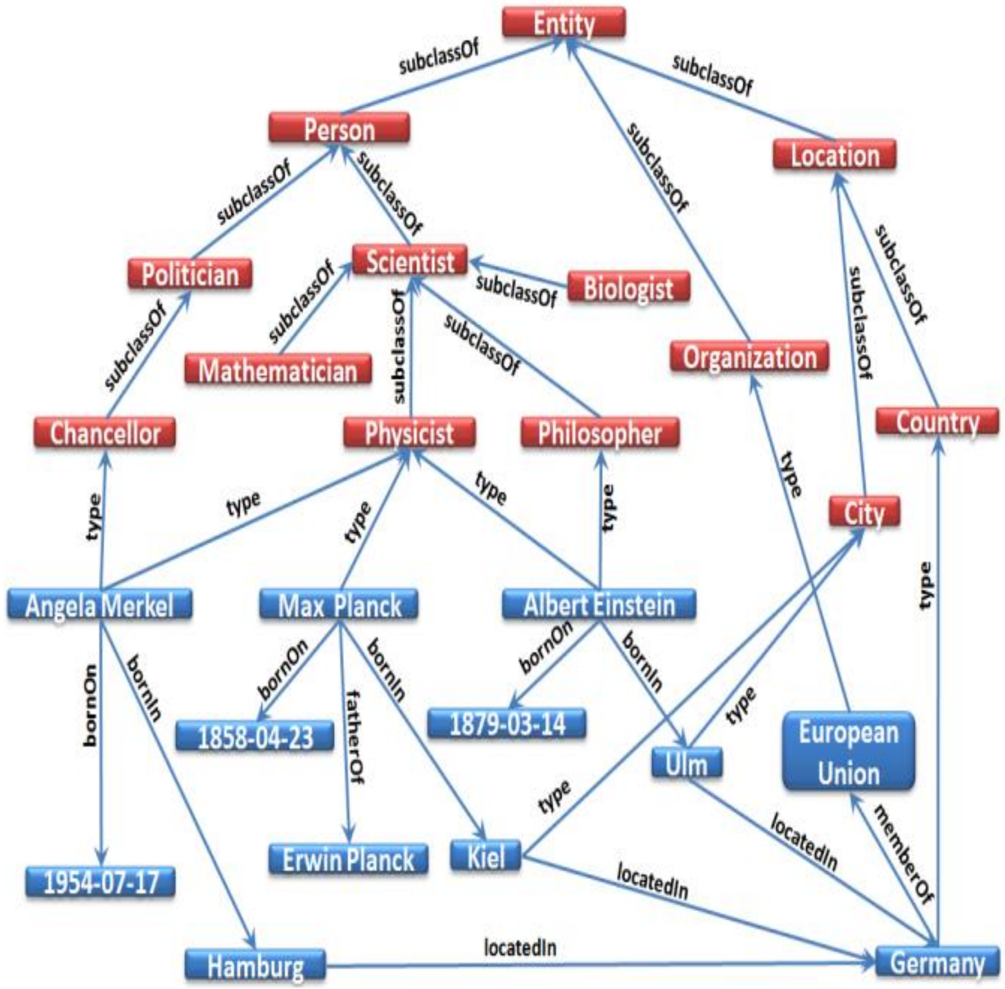
EntityAuthority (ObjectRank, PopRank, HubRank, EVA, etc.):

- define **authority transfer graph** among **entities** and **pages** with edges:
  - entity → page if entity appears in page
  - page → entity if entity is extracted from page
  - page1 → page2 if hyperlink or implicit link between pages
  - entity1 → entity2 if semantic relation between entities (from KG)
- edges can be typed and weighed by confidence and type-importance
- compared to standard Web graph, **Entity-Relationship (ER) graphs** of this kind have higher variation of edge weights

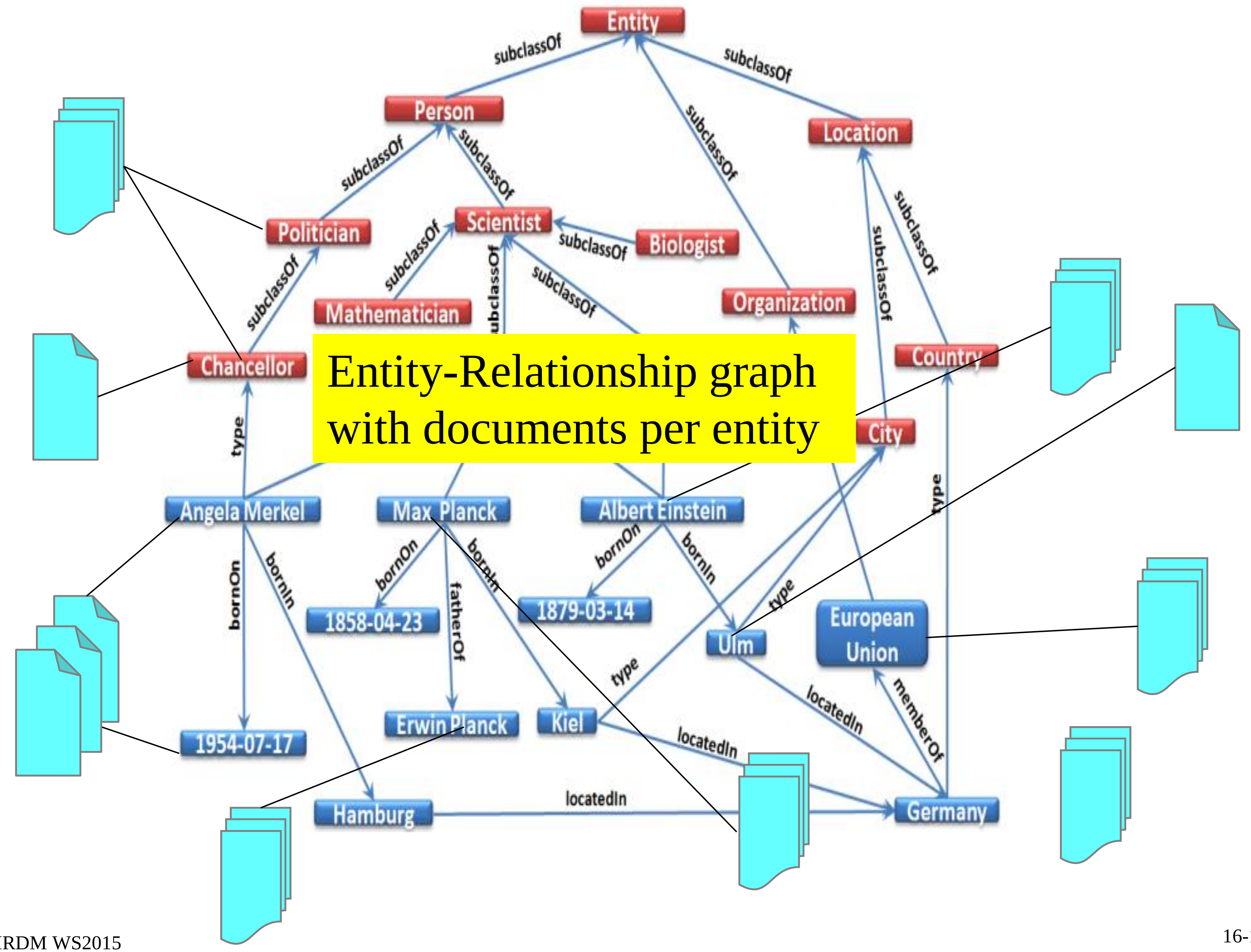
PR/HITS-style Ranking of Entities



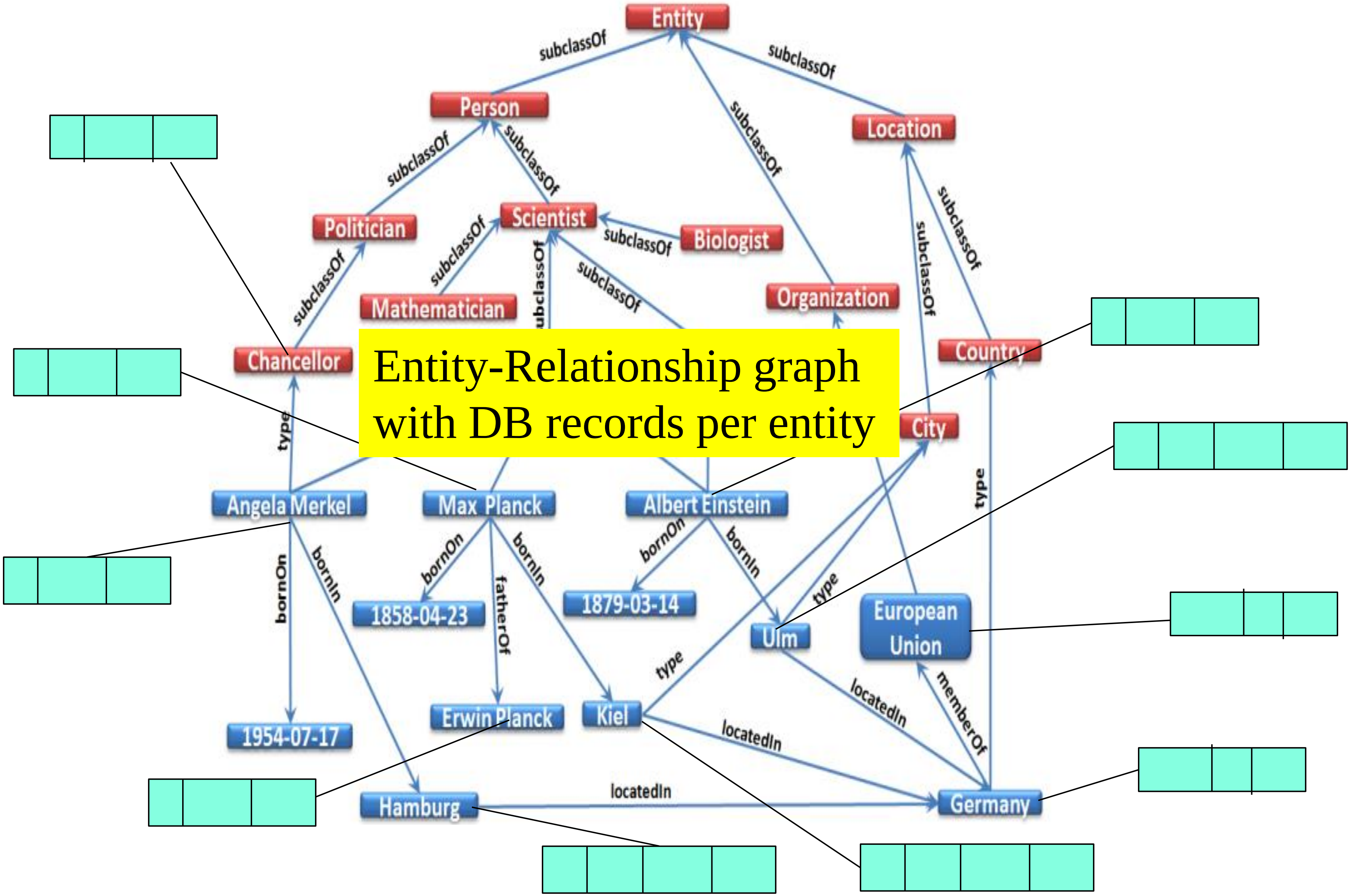
16.1.2 Entity Search with Keywords in Graph



Entity Search with Keywords in Graph



Entity Search with Keywords in Graph



Keyword Search on ER Graphs

[BANKS, Discover, DBExplorer, KUPS, SphereSearch, BLINKS, NAGA, …]

Schema-agnostic keyword search over database tables (or ER-style KG): graph of tuples with foreign-key relationships as edges

Example:

Conferences (CId, Title, Location, Year)
CPublications (PId, Title, CId)
Authors (PId, Person)
Journals (JId, Title)
JPublications (PId, Title, Vol, No, Year)
Editors (CId, Person)

Select * From * Where * Contains "*Aggarwal, Zaki, mining, knowledge*" And Year > 2005

Result is connected tree with nodes that contain as many query keywords as possible

Ranking:

$$s(tree,q) = \alpha \cdot \sum_{nodes\ n} nodeScore(n,q) + (1-\alpha) \cdot \left(1 + \sum_{edges\ e} edgeScore(e)\right)^{-1}$$

with nodeScore based on tf*idf or prob. IR
and edgeScore reflecting importance of relationships (or confidence, authority, etc.)

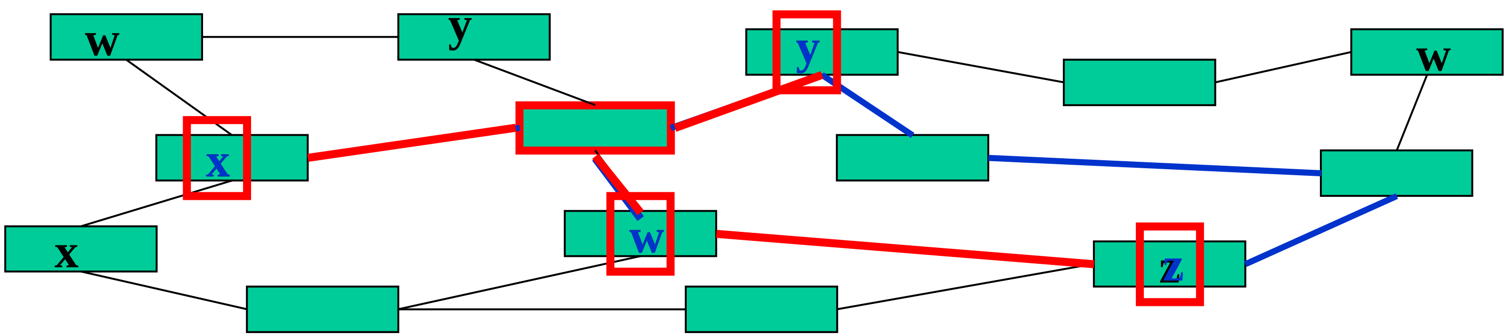
Top-k querying: compute best trees, e.g. Steiner trees (NP-hard)

Ranking by Group Steiner Trees

Answer is connected tree with nodes that contain as many query keywords as possible

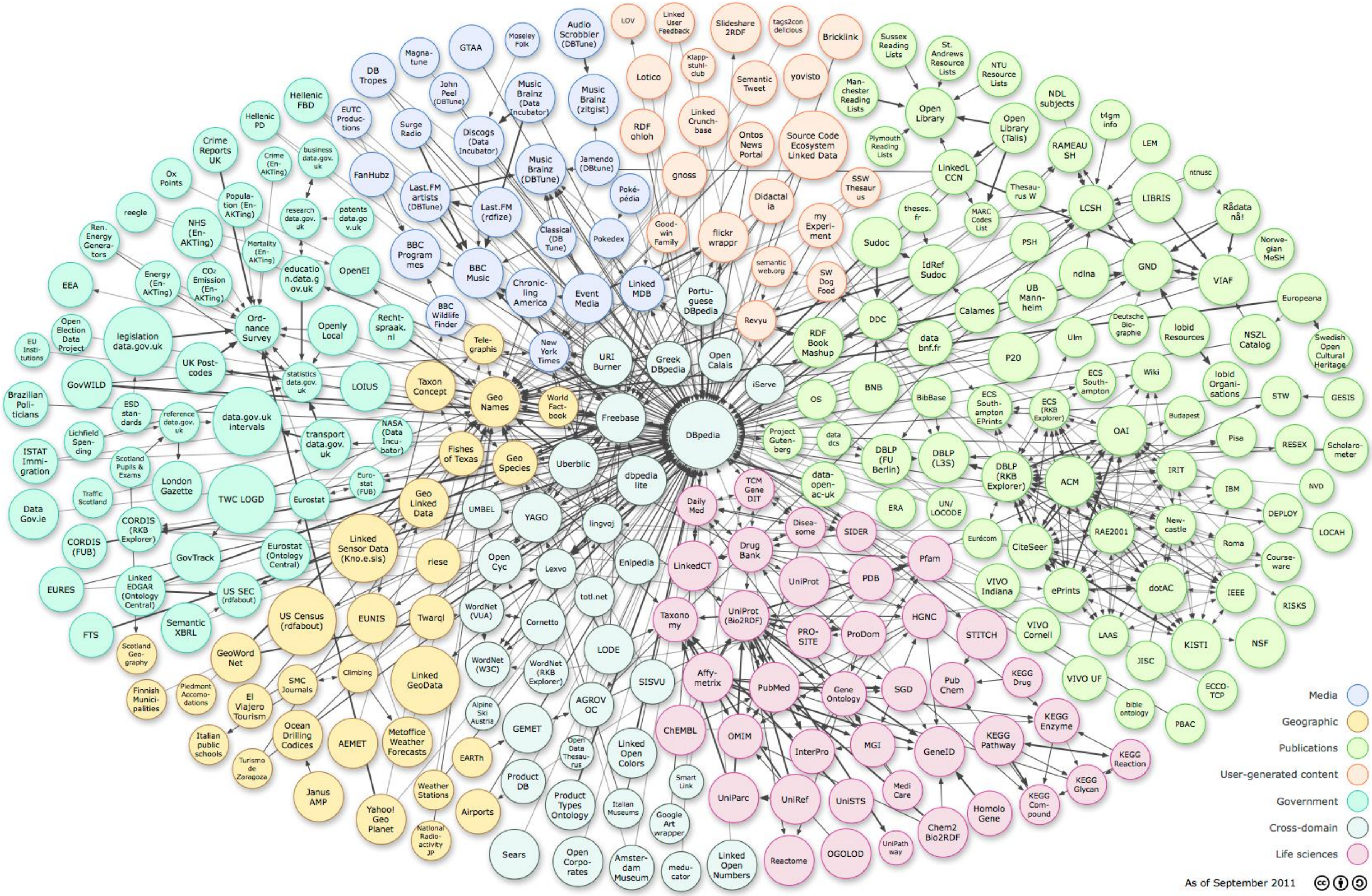
Group Steiner tree:

- match individual keywords → terminal nodes, grouped by keyword
- compute tree that connects at least one terminal node per keyword and has best total edge weight



for query: x w y z

16.1.3 Semantic Web Querying



http://richard.cyganiak.de/2007/10/lod/lod-datasets_2011-09-19_colored.png

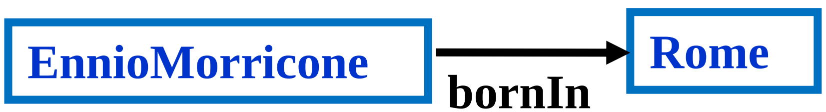
Semantic Web Data: Schema-free RDF

<u>SPO triples (statements, facts):</u>

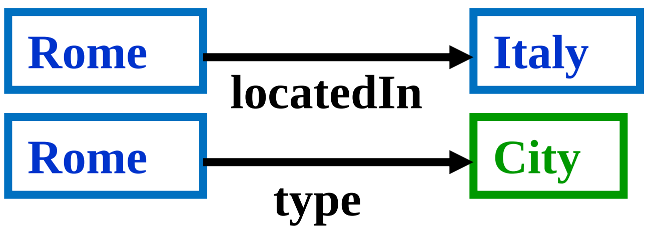
(EnnioMorricone, bornIn, Rome)
(Rome, locatedIn, Italy)
(JavierNavarrete, birthPlace, Teruel)
(Teruel, locatedIn, Spain)
(EnnioMorricone, composed, l'Arena)
(JavierNavarrete, composerOf, aTale)

(uri1, hasName, EnnioMorricone)
(uri1, bornIn, uri2)
(uri2, hasName, Rome)
(uri2, locatedIn, uri3)
…

bornIn (EnnioMorricone, Rome)



locatedIn(Rome, Italy)



- **SPO triples**: Subject – Property/Predicate – Object/Value)
- pay-as-you-go: schema-agnostic or schema later
- RDF triples form **fine-grained Entity-Relationship (ER) graph**
- popular for Linked Open Data
- open-source engines: Jena, Virtuoso, GraphDB, RDF-3X, etc.

Semantic Web Querying: SPARQL Language

Conjunctive combinations of SPO **triple patterns**
(triples with S,P,O replaced by variable(s))

Select ?p, ?c Where {
?p instanceOf Composer .
?p bornIn ?t . ?t inCountry ?c . ?c locatedIn Europe .
?p hasWon ?a .?a Name AcademyAward . }

Semantics:
return all bindings to variables that match all triple patterns
(subgraphs in RDF graph that are isomorphic to query graph)

+ filter predicates, duplicate handling, RDFS types, etc.

Select Distinct ?c Where {
?p instanceOf Composer .
?p bornIn ?t . ?t inCountry ?c . ?c locatedIn Europe .
?p hasWon ?a .?a Name ?n .
?p bornOn ?b . Filter (?b > 1945) . Filter(regex(?n, “Academy“) . }

Querying the Structured Web

flexible subgraph matching

Structure but no schema: SPARQL well suited

wildcards for properties (relaxed joins):

Select ?p, ?c Where {
?p instanceOf Composer .
?p ?r1 ?t . ?t ?r2 ?c . ?c isa Country . ?c locatedIn Europe . }

Extension: transitive paths [K. Anyanwu et al.: WWW'07]

Select ?p, ?c Where {
?p instanceOf Composer .
?p ??r ?c . ?c isa Country . ?c locatedIn Europe .
PathFilter(cost(??r) < 5) .
PathFilter (containsAny(??r,?t) . ?t isa City . }

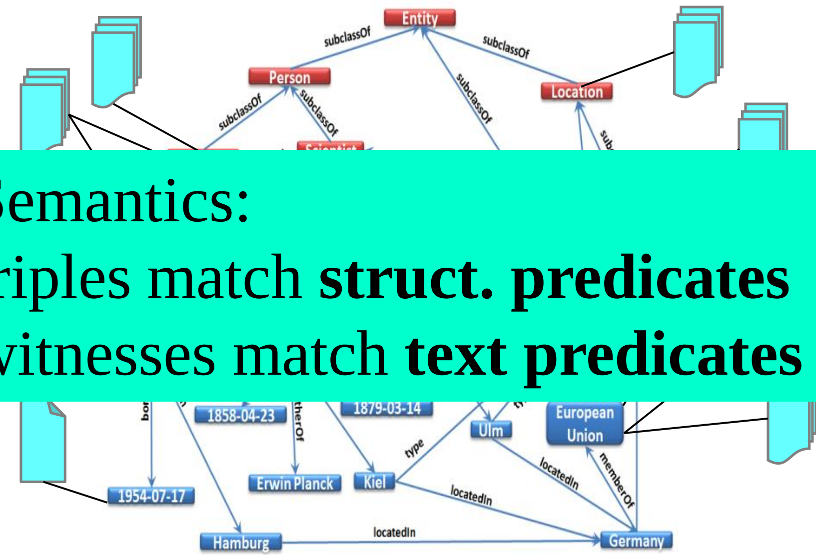
Extension: regular expressions [G. Kasneci et al.: ICDE'08]

Select ?p, ?c Where {
?p instanceOf Composer .
?p (bornIn | livesIn | citizenOf) locatedIn* Europe . }

Querying Facts & Text

Problem: not everything is in RDF

- Consider **descriptions/witnesses** of SPO facts (e.g. IE sources)
- Allow **text predicates** with each triple pattern



Semantics:
triples match **struct. predicates**
witnesses match **text predicates**

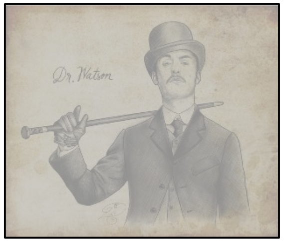
European composers who have won the Oscar, whose music appeared in dramatic western scenes, and who also wrote classical pieces ?

Select ?p Where {
?p instanceOf Composer .
?p bornIn ?t . ?t inCountry ?c . ?c locatedIn Europe .
?p hasWon ?a .?a Name AcademyAward .
?p contributedTo ?movie [western, gunfight, duel, sunset] .
?p composed ?music [classical, orchestra, cantata, opera] . }

Research issues:

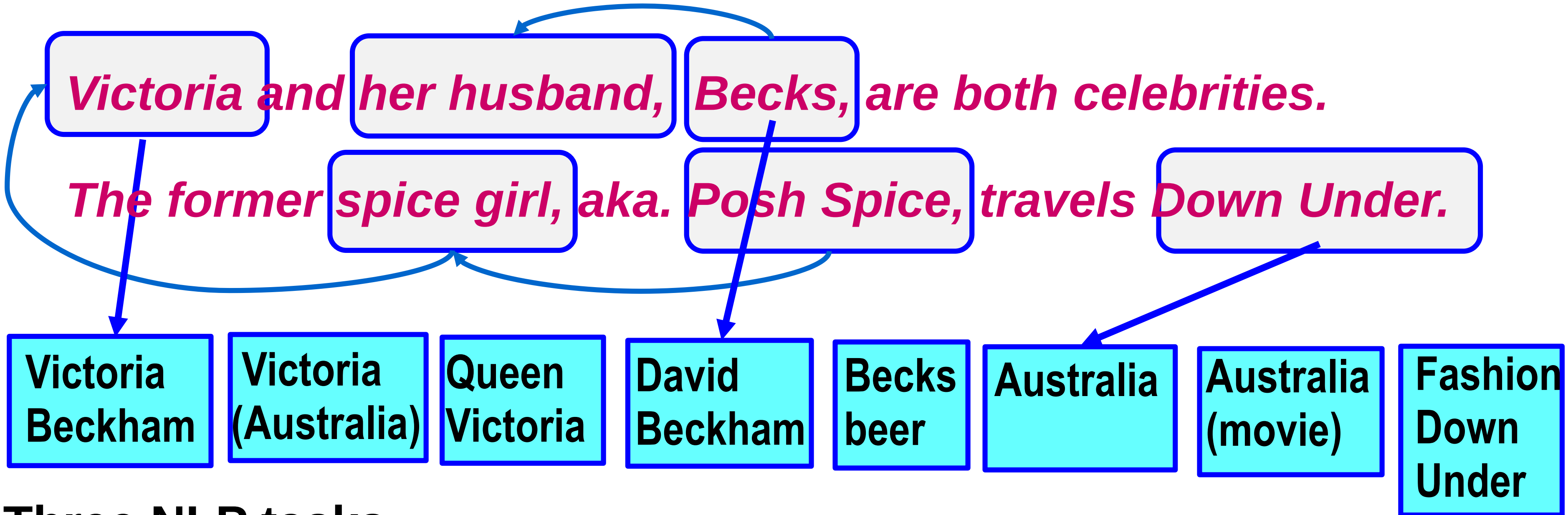
- Indexing
- Query processing
- Answer ranking

16.2 Entity Linking (aka. NERD)



Watson was better than Brad and Ken.

Named Entity Recognition & Disambiguation (NERD)



Three NLP tasks:

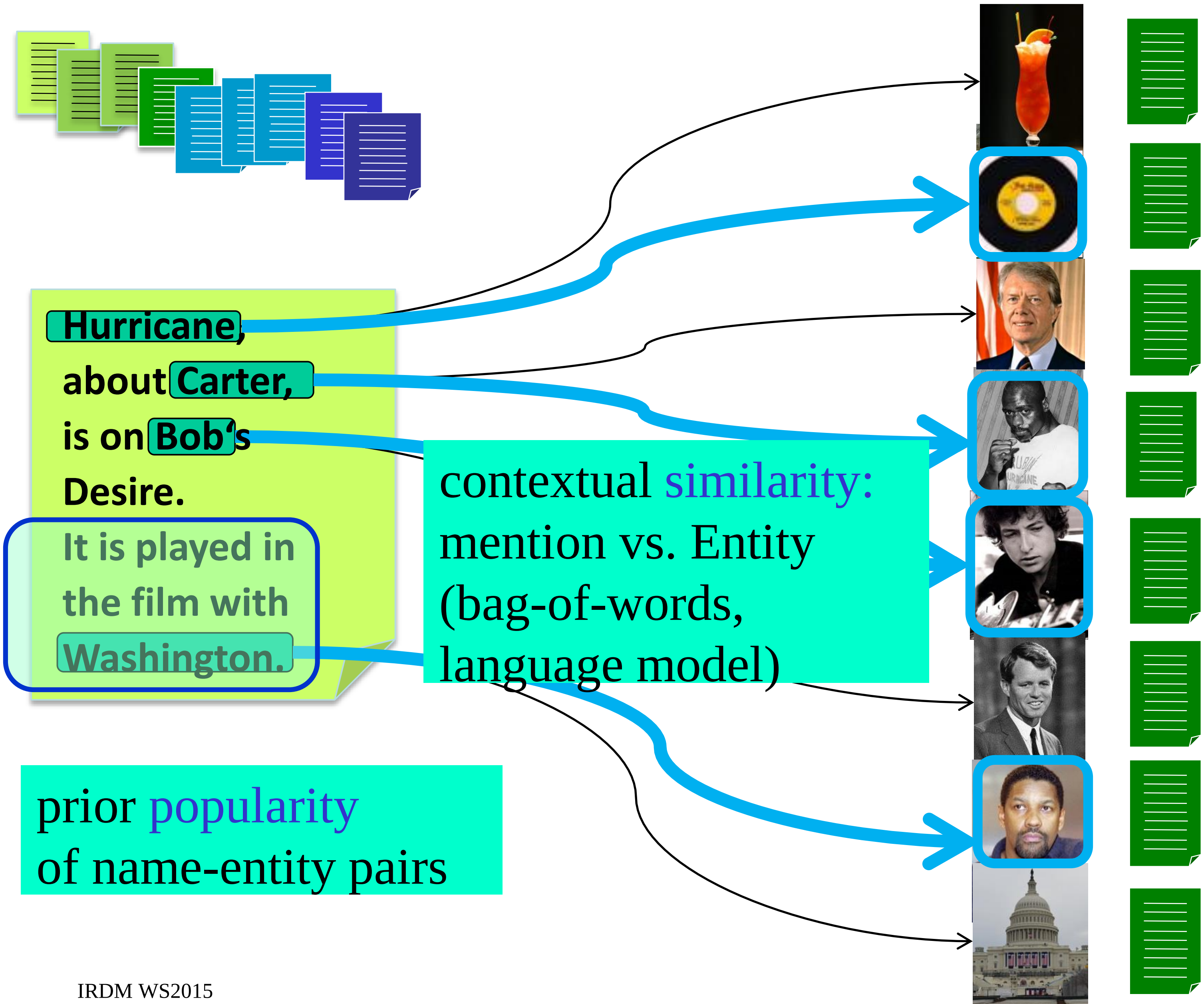
1) **named-entity detection: segment & label by HMM or CRF (e.g. Stanford NER tagger)**

2) **co-reference resolution: link to preceding NP (trained classifier over linguistic features)**

3) **named-entity disambiguation (NED): map each mention (name) to canonical entity (entry in KB)**

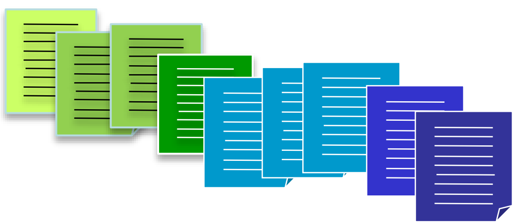
tasks 1 and 3 together: **NERD**

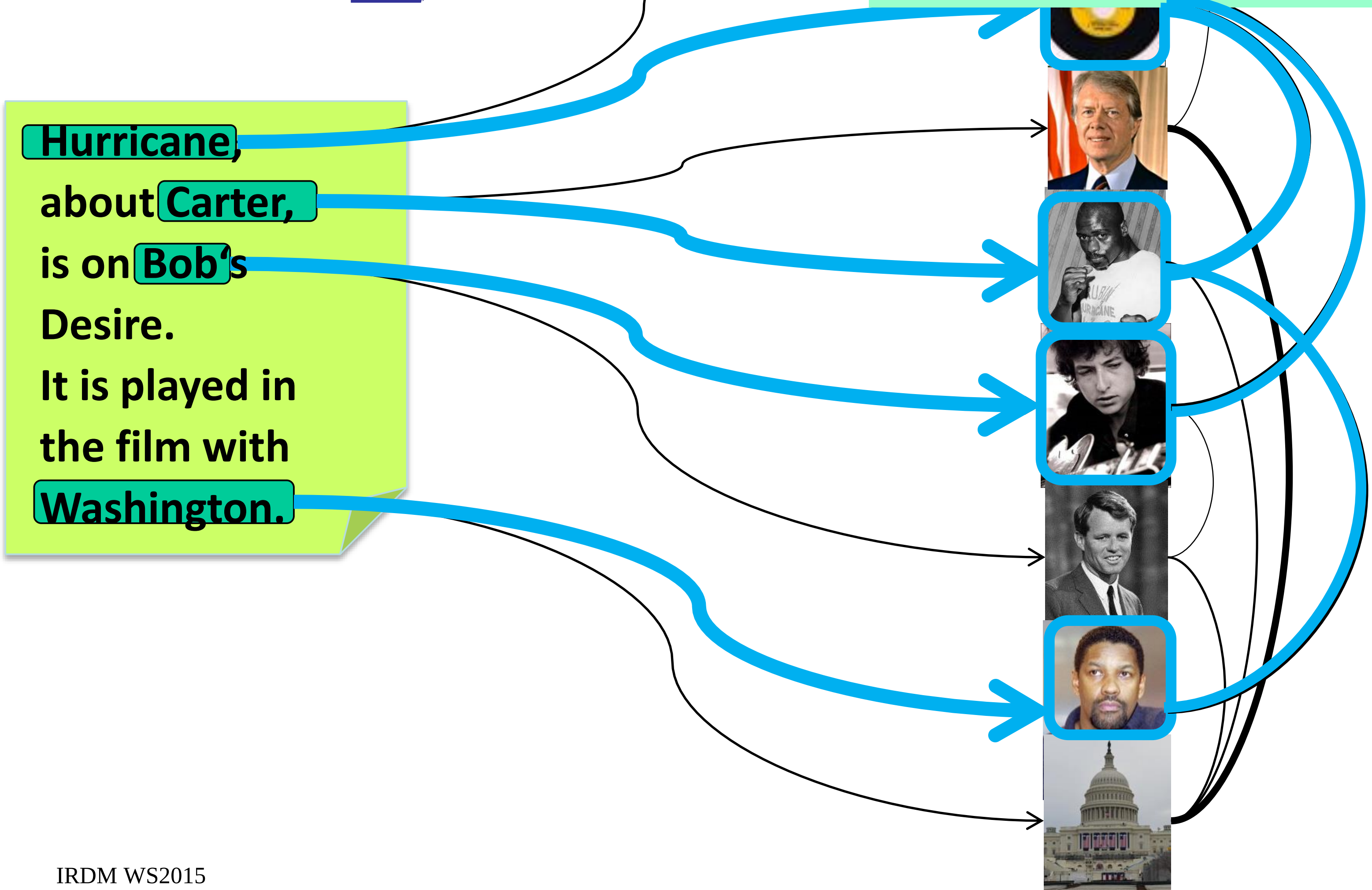
Named Entity Disambiguation (NED)



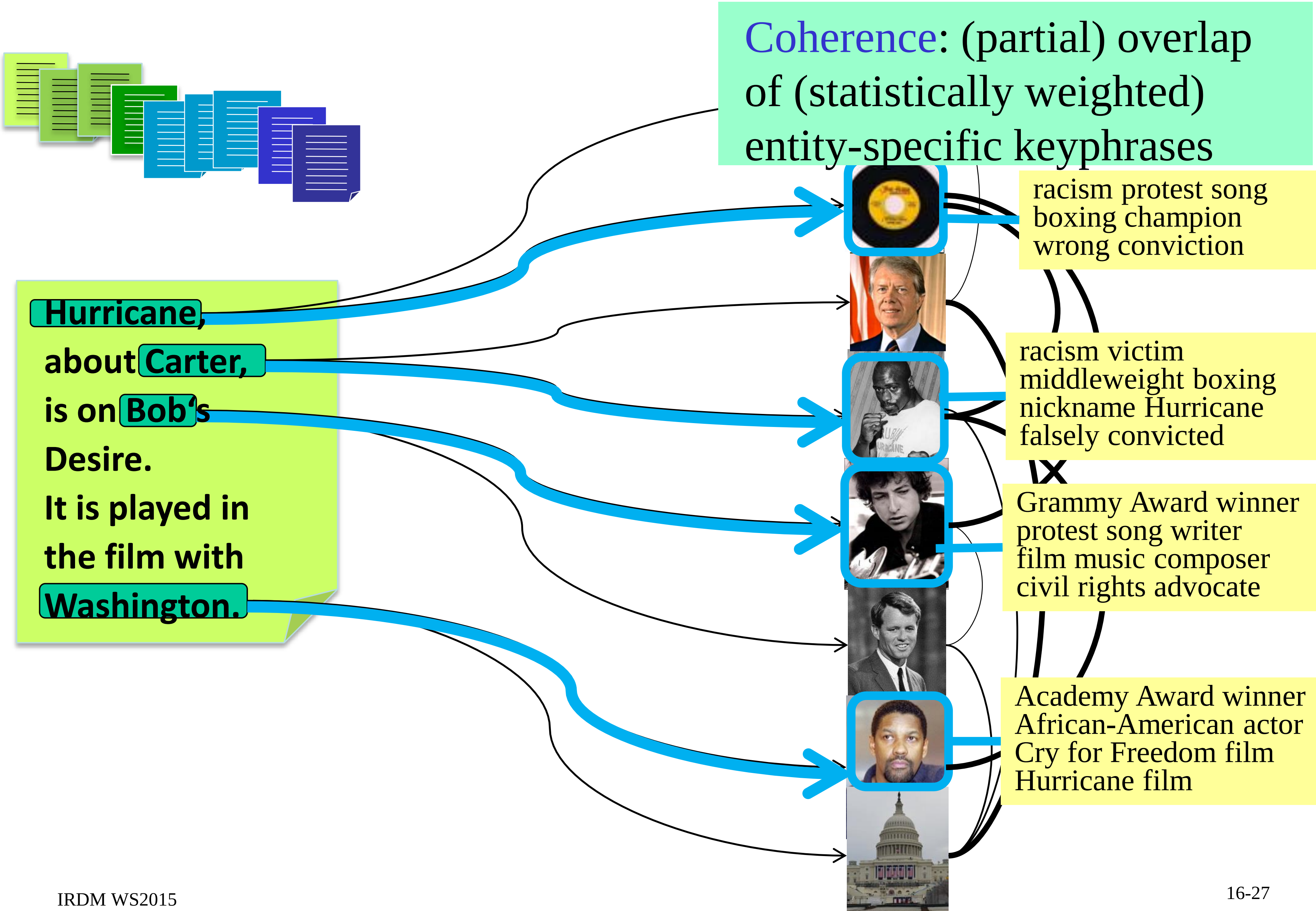
Named Entity Disambiguation (NED)

Coherence of entity pairs:
- semantic relationships
- shared types (categories)
- overlap of Wikipedia links

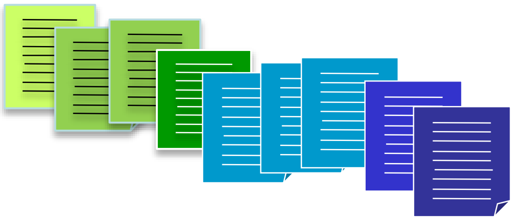




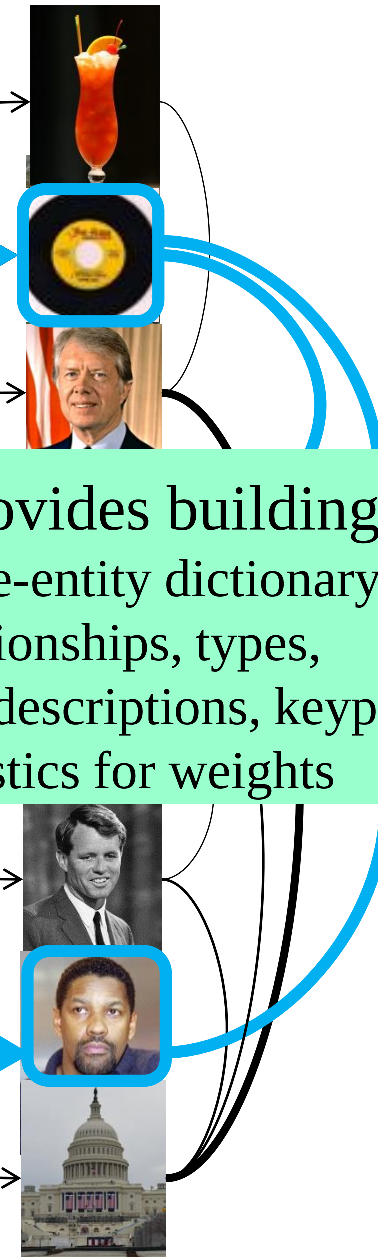
Named Entity Disambiguation (NED)



Named Entity Disambiguation (NED)



Hurricane, about Carter, is on Bob's Desire. It is played in the film with Washington.

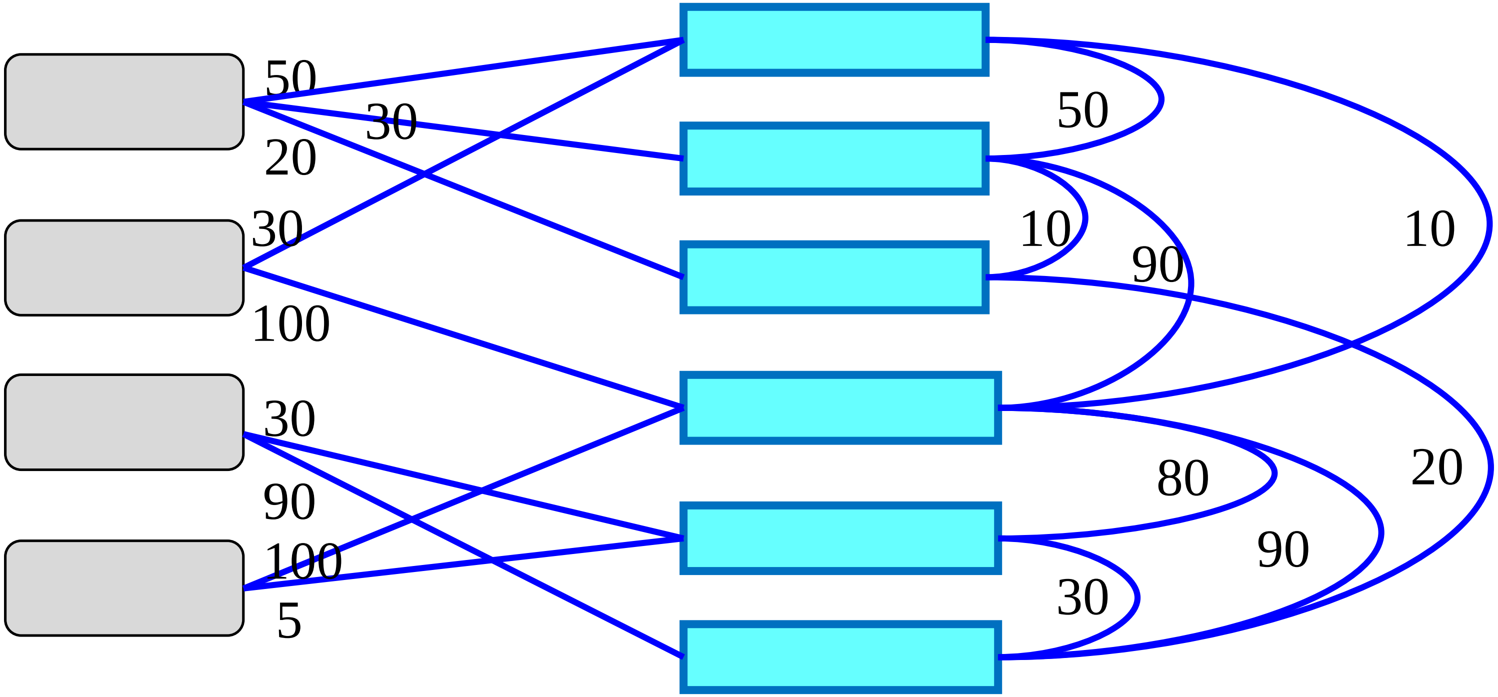


KB provides building blocks:

- name-entity dictionary,
- relationships, types,
- text descriptions, keyphrases,
- statistics for weights

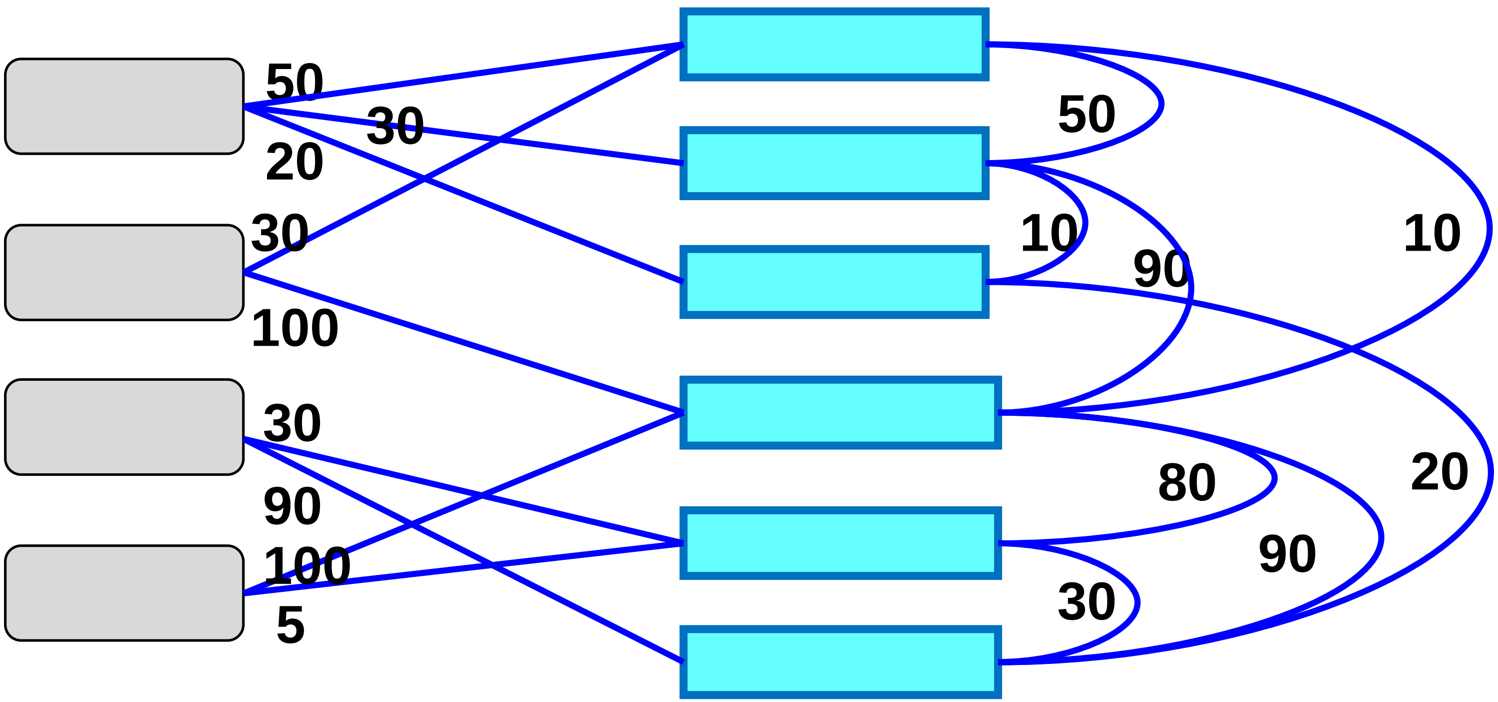
NED algorithms compute mention-to-entity mapping over weighted graph of candidates by popularity & similarity & coherence

Joint Mapping of Mentions to Entities



- Build mention-entity graph or joint-inference factor graph from knowledge and statistics in KB
- Compute high-likelihood mapping (ML or MAP) or dense subgraph such that: each m is connected to exactly one e (or at most one e)

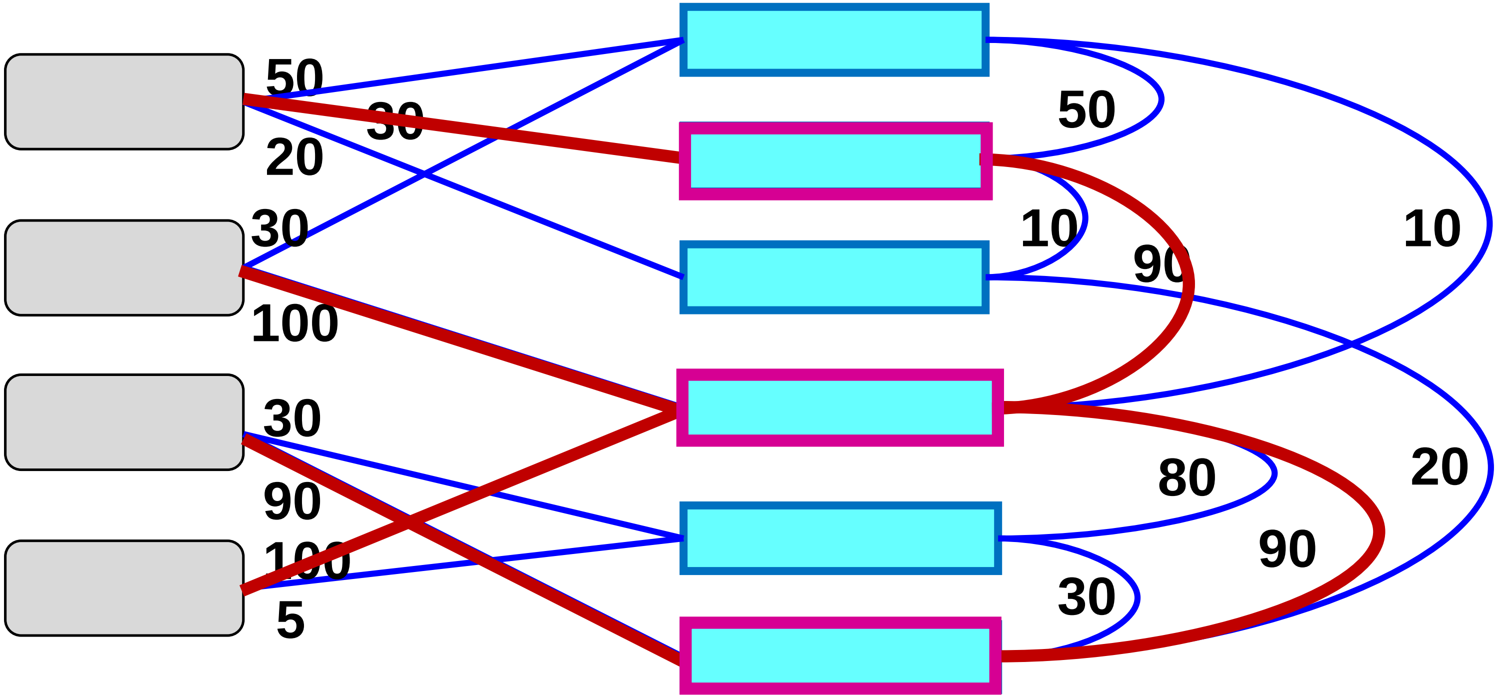
Joint Mapping: Prob. Factor Graph



Collective Learning with Probabilistic Factor Graphs
[Chakrabarti et al.: KDD'09]:

- **model P[m|e] by similarity and P[e1|e2] by coherence**
- **consider likelihood of P[m1 … mk | e1 … ek]**
- **factorize by all m-e pairs and e1-e2 pairs**
- **MAP inference: use MCMC, hill-climbing, LP etc. for solution**

Joint Mapping: Dense Subgraph



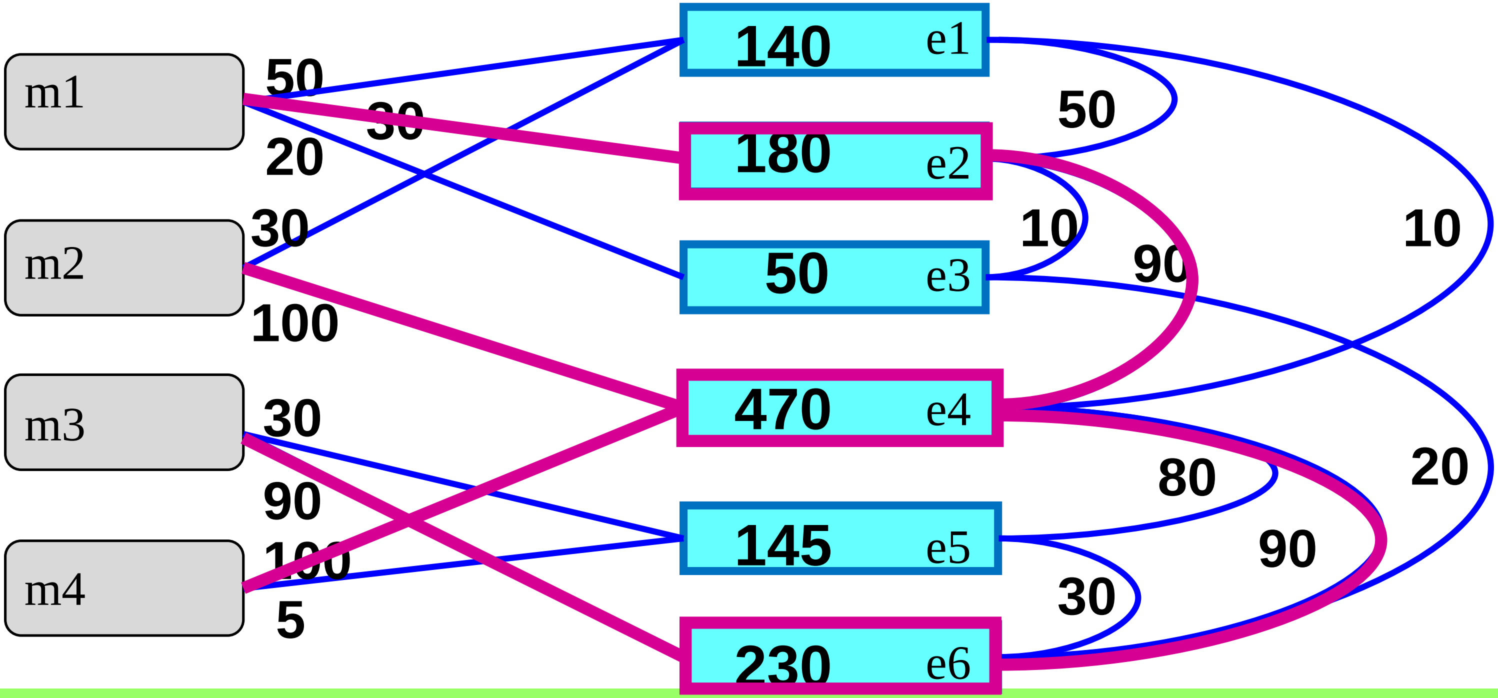
- **Compute dense subgraph such that:**

  each m is connected to exactly one e (or at most one e)
- **NP-hard → approximation algorithms**
- **Alt.: feature engineering for similarity-only method**

  [Bunescu/Pasca 2006, Cucerzan 2007,
  Milne/Witten 2008, Ferragina et al. 2010 …]

Coherence Graph Algorithm

[J. Hoffart et al.: EMNLP'11]



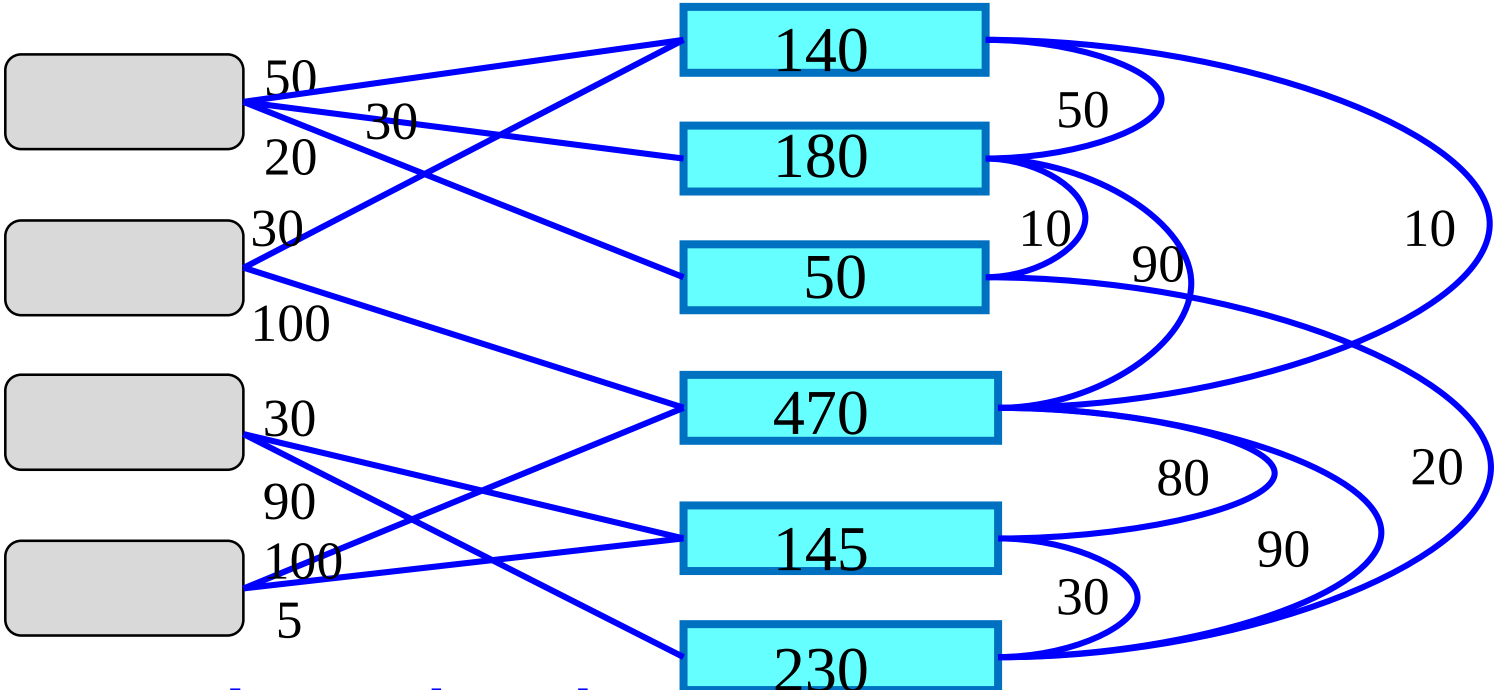
- Compute dense subgraph to maximize min weighted degree among entity nodes

  such that:

  each m is connected to exactly one e (or at most one e)
- Approx. algorithms (greedy, randomized, …), hash sketches, …
- 82% precision on CoNLL'03 benchmark
- Open-source software & online service AIDA

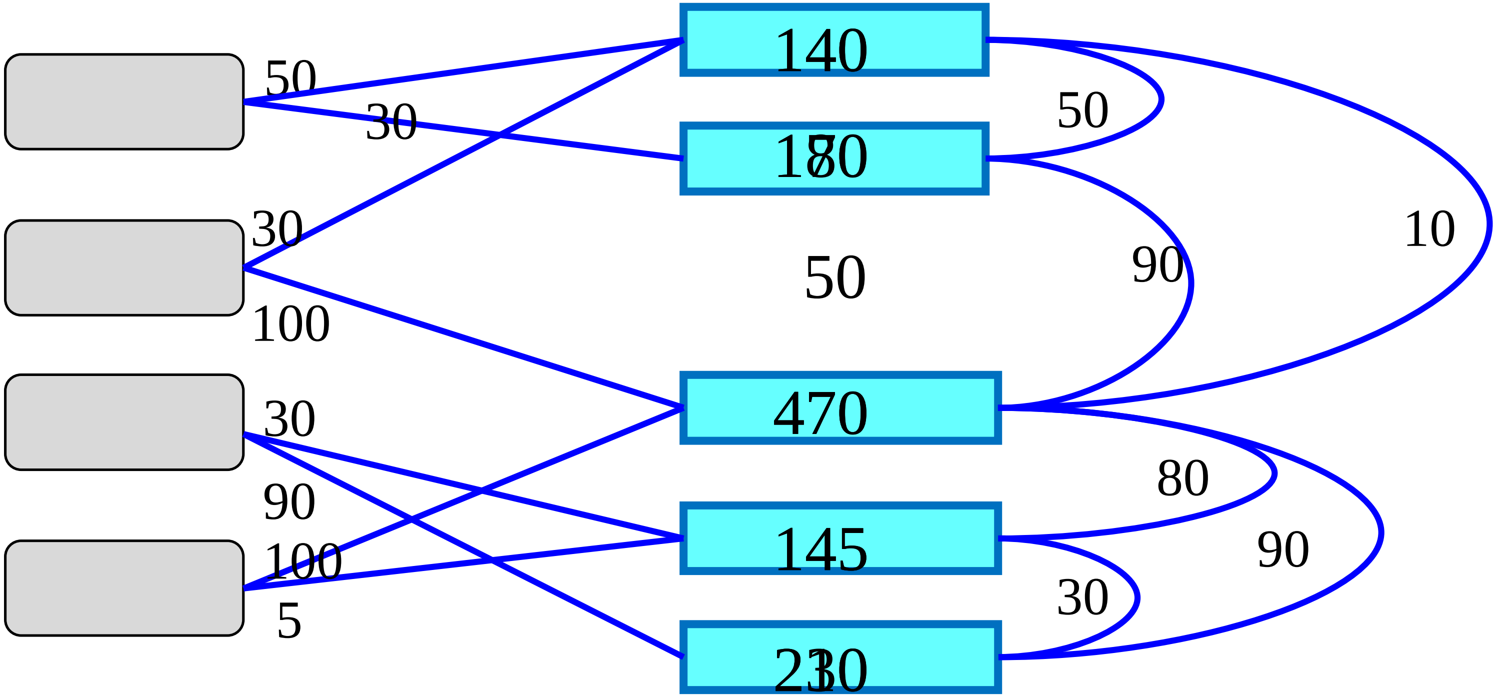
http://www.mpi-inf.mpg.de/yago-naga/aida/

Greedy Algorithm for Dense Subgraph



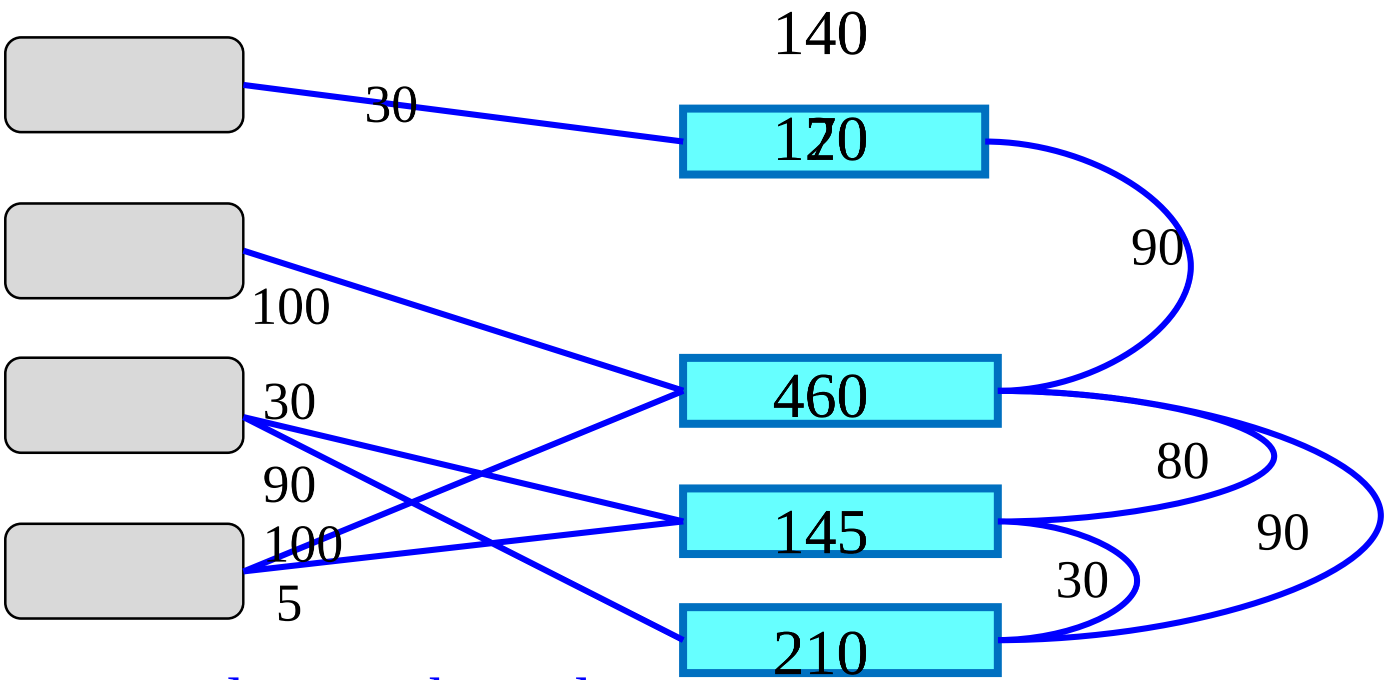
- Compute dense subgraph to
  maximize min weighted degree among entity nodes
  such that:
  each m is connected to exactly one e (or at most one e)
- Greedy approximation:
  iteratively remove weakest entity and its edges
- Keep alternative solutions, then use local/randomized search

Greedy Algorithm for Dense Subgraph



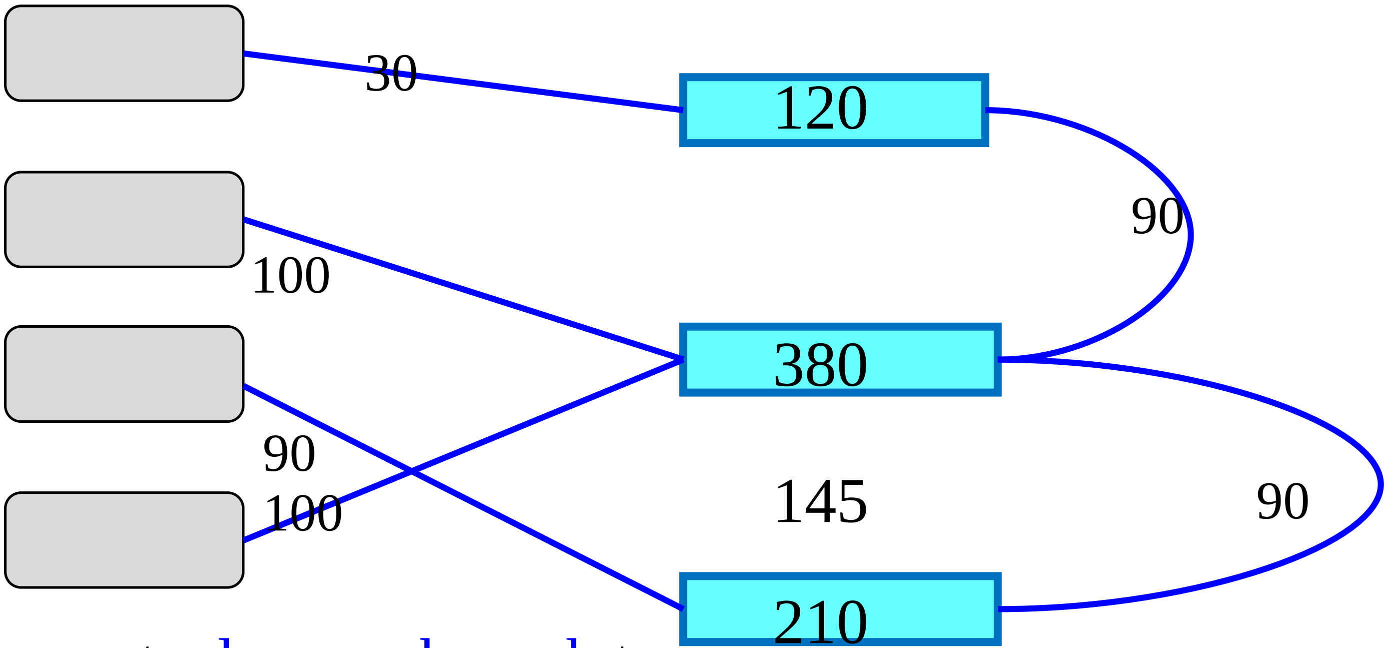
- Compute dense subgraph to
  maximize min weighted degree among entity nodes
  such that:
  each m is connected to exactly one e (or at most one e)
- Greedy approximation:
  iteratively remove weakest entity and its edges
- Keep alternative solutions, then use local/randomized search

Greedy Algorithm for Dense Subgraph



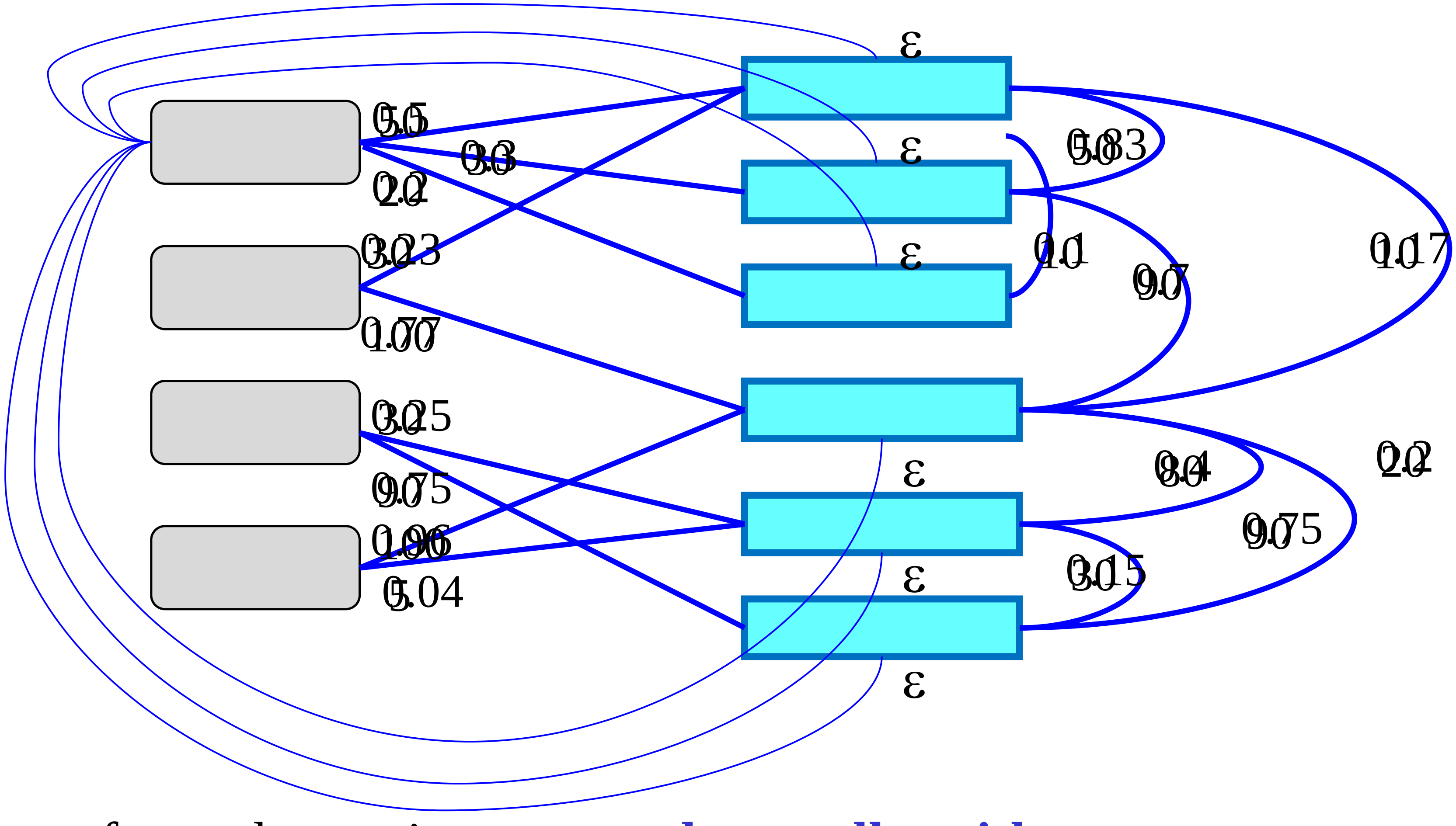
- Compute dense subgraph to
  maximize min weighted degree among entity nodes
  such that:
  each m is connected to exactly one e (or at most one e)
- Greedy approximation:
  iteratively remove weakest entity and its edges
- Keep alternative solutions, then use local/randomized search

Greedy Algorithm for Dense Subgraph



- Compute dense subgraph to
  maximize min weighted degree among entity nodes
  such that:
  each m is connected to exactly one e (or at most one e)
- Greedy approximation:
  iteratively remove weakest entity and its edges
- Keep alternative solutions, then use local/randomized search

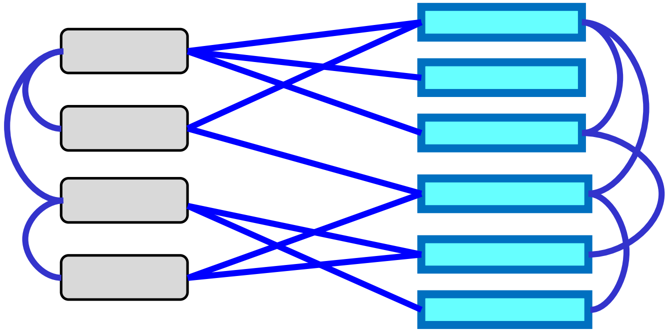
Random Walks Algorithm



- for each mention run **random walks with restart** (like Personalized PageRank with jumps to start mention(s))
- rank candidate entities by stationary visiting probability
- very efficient, decent accuracy

Integer Linear Programming

- mentions m_i
- entities e_p
- similarity $sim(cxt(m_i),cxt(e_p))$
- coherence $coh(e_p,e_q)$
- similarity $sim(cxt(m_i),cxt(m_j))$



- 0-1 decision variables: $X_{ip} = 1$ if m_i denotes e_p, 0 else

  $Z_{ij} = 1$ if m_i and m_j denote same entity

- objective function:

$$\alpha_1 \sum_{ip} sim\left(cxt(m_i), cxt(e_p)\right) X_{ip} \quad + \alpha_2 \sum_{ijpq} coh(e_p, e_q) X_{ip} X_{jq}$$

$$+ \alpha_3 \sum_{ij} sim(cxt(m_i), cxt(m_j)) Z_{iq}$$

- constraints:

  for all i,p,q: $X_{ip} + X_{iq} \leq 1$

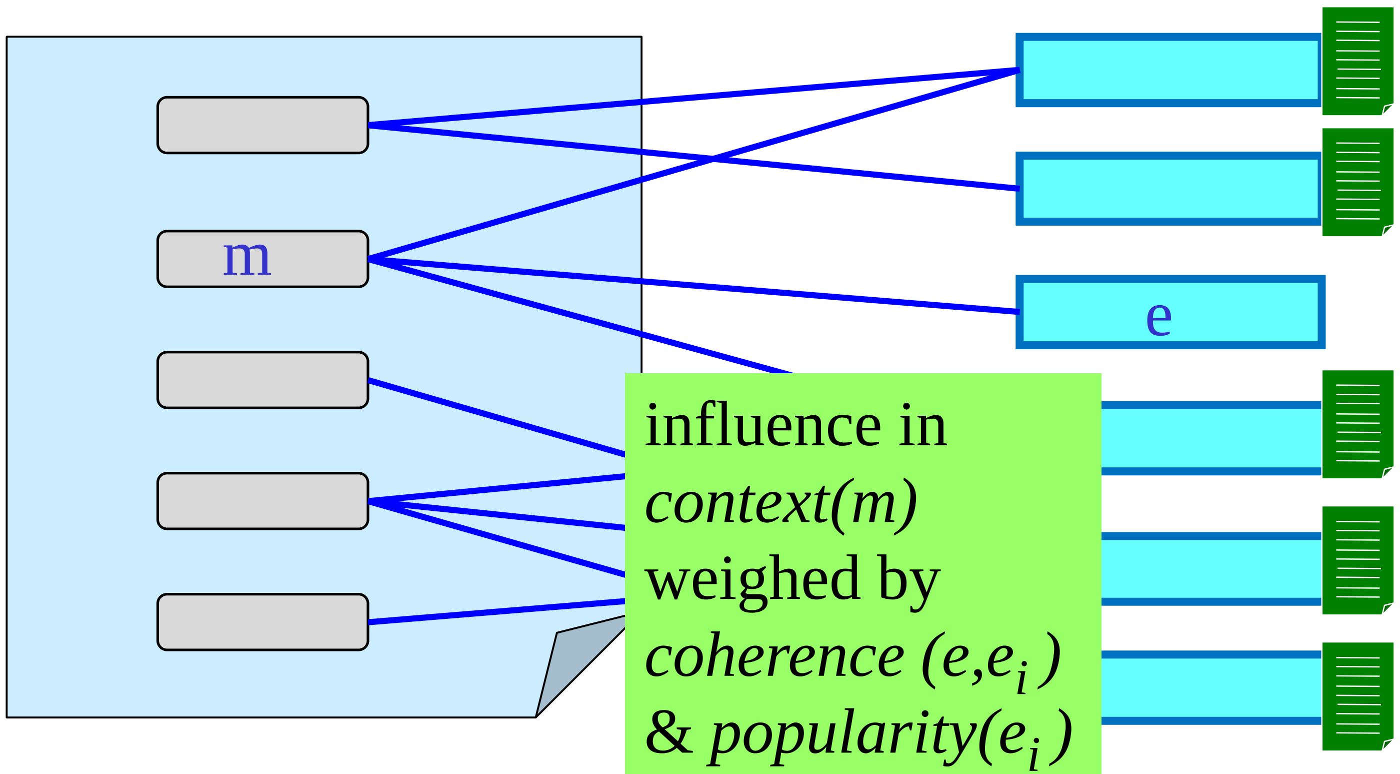
  for all i,j,p: $Z_{ij} \geq X_{ip} + X_{jp} - 1$

  for all i,j,k:
  $(1 - Z_{ij}) + (1 - Z_{jk}) \geq (1 - Z_{ik})$

  …

Coherence-aware Feature Engineering

[Cucerzan: EMNLP'07; Milne/Witten: CIKM'08, Ferragina et al.: CIKM'10]



- Avoid explicit coherence computation by turning **other mentions' candidate entities** into features
- **sim (m,e)** uses these **features in context(m)**
- special case: consider **only unambiguous mentions** or high-confidence entities (in proximity of m)

Mention-Entity Popularity Weights

[Milne/Witten 2008, Spitkovsky/Chang 2012]

- Need **dictionary** with entities‘ names:
  - full names: Arnold Alois Schwarzenegger, Los Angeles, Microsoft Corp.
  - short names: Arnold, Arnie, Mr. Schwarzenegger, New York, Microsoft, …
  - nicknames & aliases: Terminator, City of Angels, Evil Empire, …
  - acronyms: LA, UCLA, MS, MSFT
  - role names: the Austrian action hero, Californian governor, CEO of MS, …

    …

  plus gender info (useful for resolving pronouns in context):

  Bill and Melinda met at MS. They fell in love and he kissed her.

- Collect hyperlink **anchor-text / link-target** pairs from
  - Wikipedia redirects
  - Wikipedia links between articles
  - Interwiki links between Wikipedia editions
  - Web links pointing to Wikipedia articles

    …
- Build **statistics** to estimate **P[entity | name]**

Mention-Entity Similarity Edges

Precompute characteristic **keyphrases q** for each entity e:
anchor texts or noun phrases in e page with high PMI:

$$weight(q,e) = \log \frac{freq(q,e)}{freq(q)\, freq(e)}$$

"racism protest song"

Match keyphrase q of candidate e in **context** of mention m

$$score(q \mid e) \sim \frac{\#\, matching\ words}{length\ of\ cover(q)} \left(\frac{\sum_{w \in cover(q)} weight(w \mid e)}{\sum_{w \in q} weight(w \mid e)} \right)^{1+\gamma}$$

Extent of partial matches **Weight of matched words**

… and Hurricane are protest texts of songs that he wrote against racism …

Compute **overall similarity** of context(m) and candidate e

$$score(e \mid m) \sim \sum_{\substack{q \in keyphrases(e) \\ in\ context(m)}} score(q)\, dist(cover(q), m)^{-\alpha}$$

Entity-Entity Coherence Edges

Precompute **overlap of incoming links** for entities e1 and e2

$$mw\text{-}coh(e1,e2) \sim 1 - \frac{\log \max(in(e1,e2)) - \log(in(e1) \cap in(e2))}{\log |E| - \log \min(in(e1), in(e2))}$$

Alternatively compute **overlap of anchor texts** for e1 and e2

$$ngram\text{-}coh(e1,e2) \sim \frac{|ngrams(e1) \cap ngrams(e2)|}{|ngrams(e1) \cup ngrams(e2)|}$$

or **overlap of keyphrases**, or similarity of bag-of-words, or …

Optionally combine with **type distance** of e1 and e2
(e.g., Jaccard index for type instances)

For special types of e1 and e2 (locations, people, etc.)
use **spatial or temporal distance**

NERD Online Tools

J. Hoffart et al.: EMNLP 2011, VLDB 2011
http://mpi-inf.mpg.de/yago-naga/aida/
P. Ferragina, U. Scaella: CIKM 2010
http://tagme.di.unipi.it/

R. Isele, C. Bizer: VLDB 2012
http://spotlight.dbpedia.org/demo/index.html

Reuters Open Calais: http://viewer.opencalais.com/

Alchemy API: http://www.alchemyapi.com/api/demo.html

S. Kulkarni, A. Singh, G. Ramakrishnan, S. Chakrabarti: KDD 2009
http://www.cse.iitb.ac.in/soumen/doc/CSAW/

D. Milne, I. Witten: CIKM 2008
http://wikipedia-miner.cms.waikato.ac.nz/demos/annotate/

L. Ratinov, D. Roth, D. Downey, M. Anderson: ACL 2011
http://cogcomp.cs.illinois.edu/page/demo_view/Wikifier
D. Ceccarelli, C. Lucchese,S. Orlando, R. Perego, S. Trani. CIKM 2013
http://dexter.isti.cnr.it/demo/
A. Moro, A. Raganato, R. Navigli. TACL 2014
http://babelfy.org

some use Stanford NER tagger for detecting mentions
http://nlp.stanford.edu/software/CRF-NER.shtml

NERD at Work

https://gate.d5.mpi-inf.mpg.de/webaida/

Hurricane, a protest song about Carter, is on Bob's Desire.
Scarlet plays the violin on this piece. In the movie, Washington plays the boxer.

Disambiguate

Input Type:TEXT **Overall runtime:**33 sec(s)

Hurricane [Hurricane (Bob Dylan song)], a protest song about **Carter** [Rubin Carter], is on **Bob** [Bob Dylan]'s **Desire** [Desire (Bob Dylan album)]. **Scarlet** [Scarlet Rivera] plays the violin on this piece. In the movie, **Washington** [Denzel Washington] plays the boxer.

Run Information | Graph | Removal Steps

- 0: Hurricane
- 32: Carter
- 46: Bob
- 52: Desire
- 62: Scarlet
- 116: Washington

NERD at Work

https://gate.d5.mpi-inf.mpg.de/webaida/

Disambiguation Method:

prior | prior+sim | prior+sim+coherence

Parameters

Prior-Similarity-Coherence balancing ratio:

prior VS. sim. balance = 0.13 **(prior+sim.) VS. coh.** balance 0.71

Ambiguity degree 10

Coherence robustness test threshold: 0.9

Entities Type Filters:

Mention Extraction:

Stanford NER | Manual

You can manually tag the mentions by putting them between [[and]]. HTML Tables are automatcially disambiguated in the manual mode.

Fast Mode:

Enabled

Hurricane, a protest song about Carter, is on Bob's Desire.
Scarlet plays the violin on this piece. In the movie, Washington plays the boxer.

Disambiguate

Input Type:TEXT **Overall runtime:**33 sec(s)

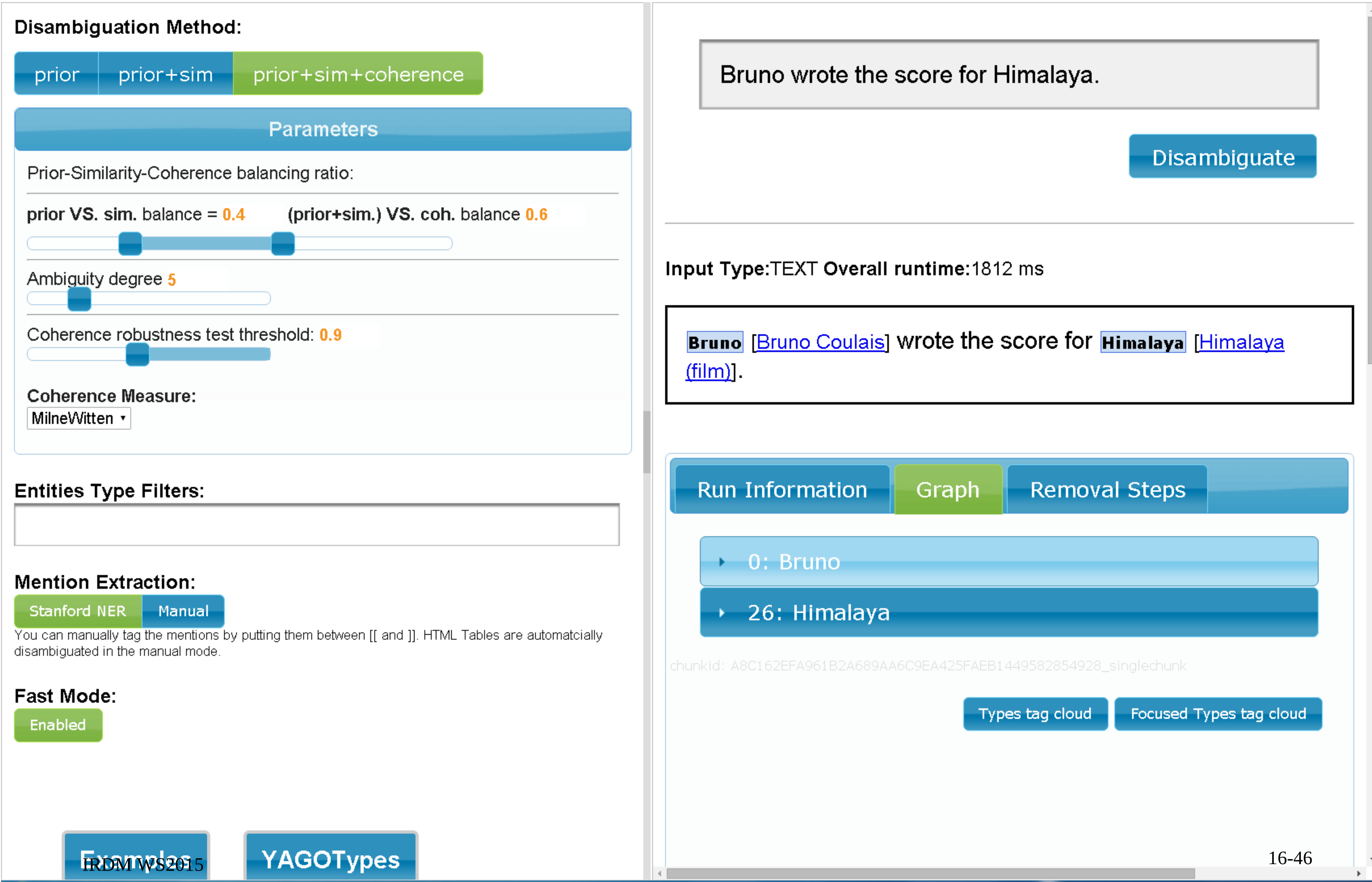
Hurricane [Hurricane (Bob Dylan song)], a protest song about **Carter** [Rubin Carter], is on **Bob** [Bob Dylan]'s **Desire** [Desire (Bob Dylan album)]. **Scarlet** [Scarlet Rivera] plays the violin on this piece. In the movie, **Washington** [Denzel Washington] plays the boxer.

32: Carter

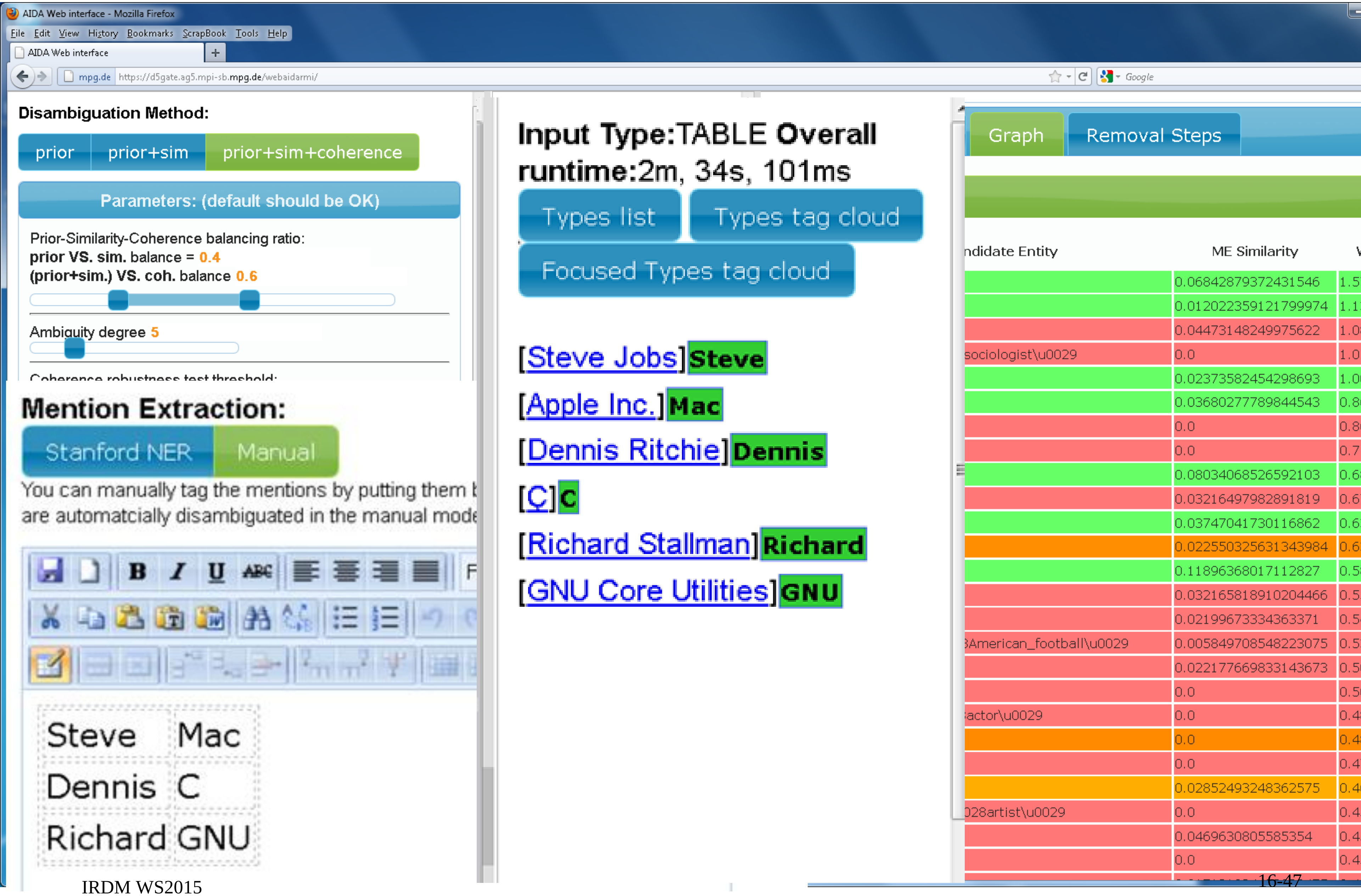
	Candidate Entity	ME Similarity	Weighted Degree	Weighted Degree when removed/final	Connected Entities
Info	Rubin Carter	0.007440300887298156	0.3672384453830128	0.017696436920930227	199 Show
Info	Joe Carter	0.0	0.3281050927116556	0.3281050927116556	188 Show
Info	Jimmy Carter	0.01103638778377025	0.3025790075617965	0.013351114882815256	320 Show
Info	Gary Carter	0.0021657937926300736	0.27194405292066054	0.27194405292066054	159 Show
Info	Paul Carter (baseball)	0.0	0.19680276201878621	0.19680276201878621	87 Show
Info	Vince Carter	4.1435682855787666E-4	0.1281591894396449	0.1281591894396449	88 Show
Info	Jay-Z	0.00730218654460134	0.12814442111832083	0.011735882716700024	137 Show
Info	Carter Elliott	0.0	0.1118463610679272	0.1118463610679272	47 Show
Info	Lance Carter	0.0	0.110008842052524	0.110008842052524	55 Show
Info	Steve Carter (baseball)	0.0	0.1005279520503617	0.1005279520503617	46 Show
Info	Chris Carter (right-handed hitter)	0.0	0.09913125899246221	0.09913125899246221	50 Show
Info	Arnold Carter	0.0	0.09623832488634608	0.09623832488634608	42 Show
Info	Howie Carter	0.0	0.09575478704689618	0.09575478704689618	40 Show
Info	Chris Carter (left-handed hitter)	3.774760610665208E-4	0.09537978696432067	0.09537978696432067	45 Show
Info	Nick Carter (baseball)	0.0	0.09167177180852937	0.09167177180852937	39 Show
Info	Sol Carter	0.0	0.09135182831121434	0.09135182831121434	38 Show
Info	Helena Bonham Carter	8.590379156735183E-4	0.09124507304617609	0.09124507304617609	68 Show
Info	Benny Carter	0.001310040883999477	0.09089849194529637	0.09089849194529637	67 Show
Info	Jeff Carter (pitcher)	0.0	0.09074559389855853	0.09074559389855853	40 Show
Info	Anthony Carter (American football)	4.080916063142848E-4	0.08487224122114082	0.08487224122114082	50 Show
Info	Ron Carter	0.006379385398268004	0.08444139387442567	0.010422108122627302	67 Show

NERD at Work

https://gate.d5.mpi-inf.mpg.de/webaida/



NERD on Tables



General Word Sense Disambiguation (WSD)

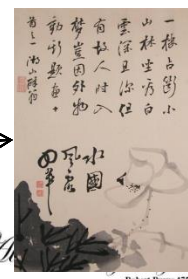
Which **song writers** **covered** **ballads** **written by** the **Stones** ?

{songwriter, composer}

{cover, perform}

{cover, report, treat}

{cover, help out}



Verb

- S: (v) cover (provide with a covering or cause to be covered ... *handkerchief"; "cover the child with a blanket"; "cover the g*
- S: (v) cover, spread over (form a cover over) *"The grass cov*
- S: (v) cover, continue, extend (span an interval of distance, s... *extended over five years"; "The period covered the turn of ... extends over the hills on the horizon"; "This farm covers so... Archipelago continues for another 500 miles"*
- S: (v) cover (provide for) *"The grant doesn't cover my salary*
- S: (v) cover, treat, handle, plow, deal, address (act on verbal... expression) *"This book deals with incest"; "The course cov... Civilization"; "The new book treats the history of China"*
- S: (v) embrace, encompass, comprehend, cover (include in ... something broader; have as one's sphere or territory) *"This ... wide range of people from different backgrounds"; "this sho... group"*
- S: (v) traverse, track, cover, cross, pass over, get over, get a... across (travel across or pass over) *"The caravan covered a*
- S: (v) report, cover (be responsible for reporting the details o... *reported on China in the 1950's"; "The cub reporter covere*
- S: (v) cover (hold within range of an aimed firearm)
- S: (v) cover (to take an action to protect against future proble... *the drawer twice just to cover yourself"*
- S: (v) cover, cover up (hide from view or knowledge) *"The P... that he bugged the offices in the White House"*
- S: (v) cover (protect or defend (a position in a game)) *"he co*
- S: (v) cover (maintain a check on; especially by patrolling) *"T... the top floor"*
- S: (v) cover, insure, underwrite (protect by insurance) *"The in*
- S: (v) cover, compensate, overcompensate (make up for sho... inferiority by exaggerating good qualities) *"he is compensa*
- S: (v) cover (invest with a large or excessive amount of some... *herself with glory"*
- S: (v) cover (help out by taking someone's place and tempor... responsibilities) *"She is covering for our secretary who is ill*
- S: (v) cover (be sufficient to meet, defray, or offset the charge... *to cover the check?"*
- S: (v) cover (spread over a surface to conceal or protect) *"T*
- S: (v) shroud, enshroud, hide, cover (cover as if with a shrou... *civilization are shrouded in mystery"*
- S: (v) breed, cover (copulate with a female, used especially ... *covers the mare"*
- S: (v) overlay, cover (put something on top of something else... *of gravy"*
- S: (v) cover (play a higher card than the one previously playe...
- S: (v) cover (be responsible for guarding an opponent in a g...
- S: (v) brood, hatch, cover, incubate (sit on (eggs)) *"Birds bro... the eggs"*
- S: (v) cover, wrap up (clothe, as if for protection from the ele...

NERD Challenges

High-throughput NERD: semantic indexing

Low-latency NERD: speed-reading

popular vs. long-tail entities, general vs.specific domain

Short and difficult texts:

queries – example: "Borussia victory over Bayern"

tweets, headlines, etc.

fictional texts: novels, song lyrics, TV sitcoms, etc.

Handle **long-tail** and **newly emerging entities**

General WSD for classes, relations, general concepts

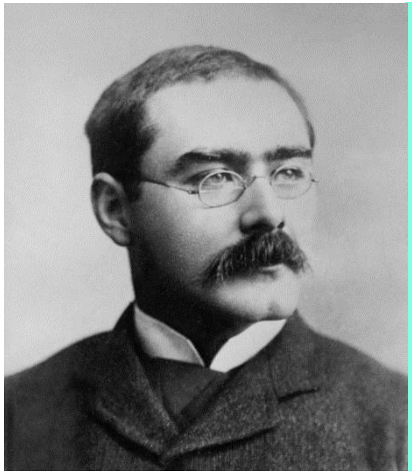
for Web tables, lists, questions, dialogs, summarization, …

Leverage **deep-parsing** features & **semantic typing**

example: ***Page played Kashmir on his Gibson***

subj obj mod

16.3 Natural Language Question Answering

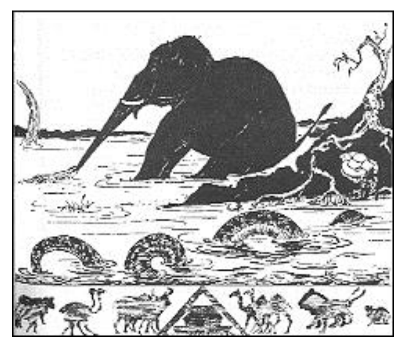


Six honest men

I have six honest serving men
They taught me all I knew.
Their names are ***What*** *and* ***Where*** *and* ***When***
and ***Why*** *and* ***How*** *and* ***Who***.

Rudyard Kipling
(1865-1936)

from „The Elephant‘s Child“ (1900)



Question Answering (QA)

Different kinds of questions:

- **Factoid questions:**
  Where is the Louvre located?
  Which metro line goes to the Louvre?
  Who composed Knockin' on Heaven's Door?
  Which is the highest waterfall on Iceland?
- **List questions:**
  Which museums are there in Paris?
  Which love songs did Bob Dylan write
  Which impressive waterfalls does Iceland have?
- **Relationship questions:**
  Which Bob Dylan songs were used in movies?
  Who covered Bob Dylan? Who performed songs written by Bob Dylan?
- **How-to questions:**
  How do I get from Paris Est to the Louvre?
  How do I stop pop-up ads in Mozilla?
  How do I cross a turbulent river on a wilderness hike?

QA System Architecture

1 **Classify question**: Who, When, Where, …

Where is the Louvre located?

2 **Generate web query/queries**: informative phrases (with expansion)

Louvre; Louvre location; Louvre address;

3 **Retrieve passages**: short (var-length) text snippets from results

… The Louvre Museum is at Musée du Louvre, 75058 Paris Cedex 01 …

… The Louvre is located not far from the Seine. The Seine divides Paris …

… The Louvre is in the heart of Paris. It is the most impressive museum …

… The Louvre can only be compared to the Eremitage in St. Petersburg …

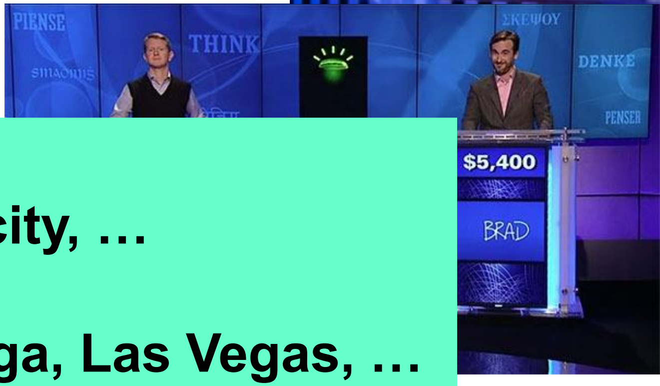
4 **Extract candidate answers** (e.g. noun phrases near query words)

Musée du Louvre, Seine, Paris, St. Petersburg, museum, …

5 **Aggregate candidates** over all passages

6 **Rank candidates**: using passage LM‘s

Deep Question Answering



This town is known as "Sin City" & its
downtown is "Glitter Gulch"

Q: Sin City ?
→ movie, graphical novel, nickname for city, …
A: Vegas ? Strip ?
→ Vega (star), Suzanne Vega, Vincent Vega, Las Vegas, …
→ comic strip, striptease, Las Vegas Strip, …

This American city has two airports
named after a war hero and a WW II battle

question classification & decomposition → **knowledge back-ends**

D. Ferrucci et al.: Building Watson. AI Magazine, Fall 2010.
IBM Journal of R&D 56(3/4), 2012: This is Watson.

More Jeopardy! Questions

24-Dec-2014: http://www.j-archive.com/showgame.php?game_id=4761
Categories: Alexander the Great, Santa's Reindeer Party,
Making Some Coin, TV Roommates, The „NFL"

- **Alexander the Great was born in 356 B.C. to King Philip II & Queen Olympias of this kingdom**

  (Macedonia)
- **Against an Indian army in 326 B.C., Alexander faced these beasts, including the one ridden by King Porus**

  (elephants)
- **In 2000 this Shoshone woman first graced our golden dollar coin**

  (Sacagawea)
- **When her retirement home burned down in this series, Sophia moved in with her daughter Dorothy and Rose & Blanche**

  (The Golden Girls)
- **Double-winged "mythical" insect**

  (dragonfly)

Difficult of Jeopardy! Questions

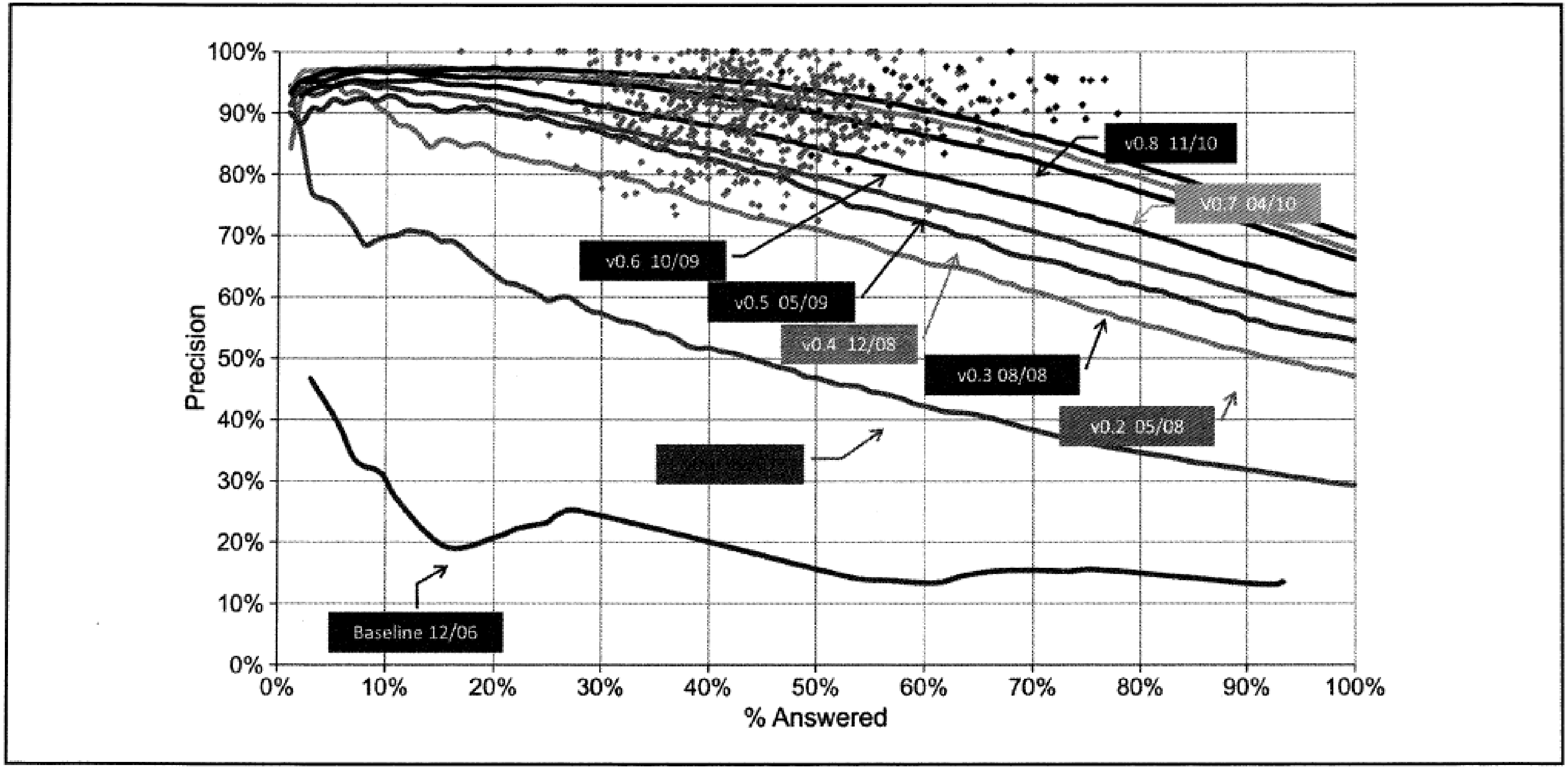


Figure 2

Incremental progress in answering precision on the Jeopardy! challenge: June 2007 to November 2011.

Source: IBM Journal of R&D 56(3-4), 2012

Question Analysis

Train a classifier for the semantic answer type
and process questions by their type

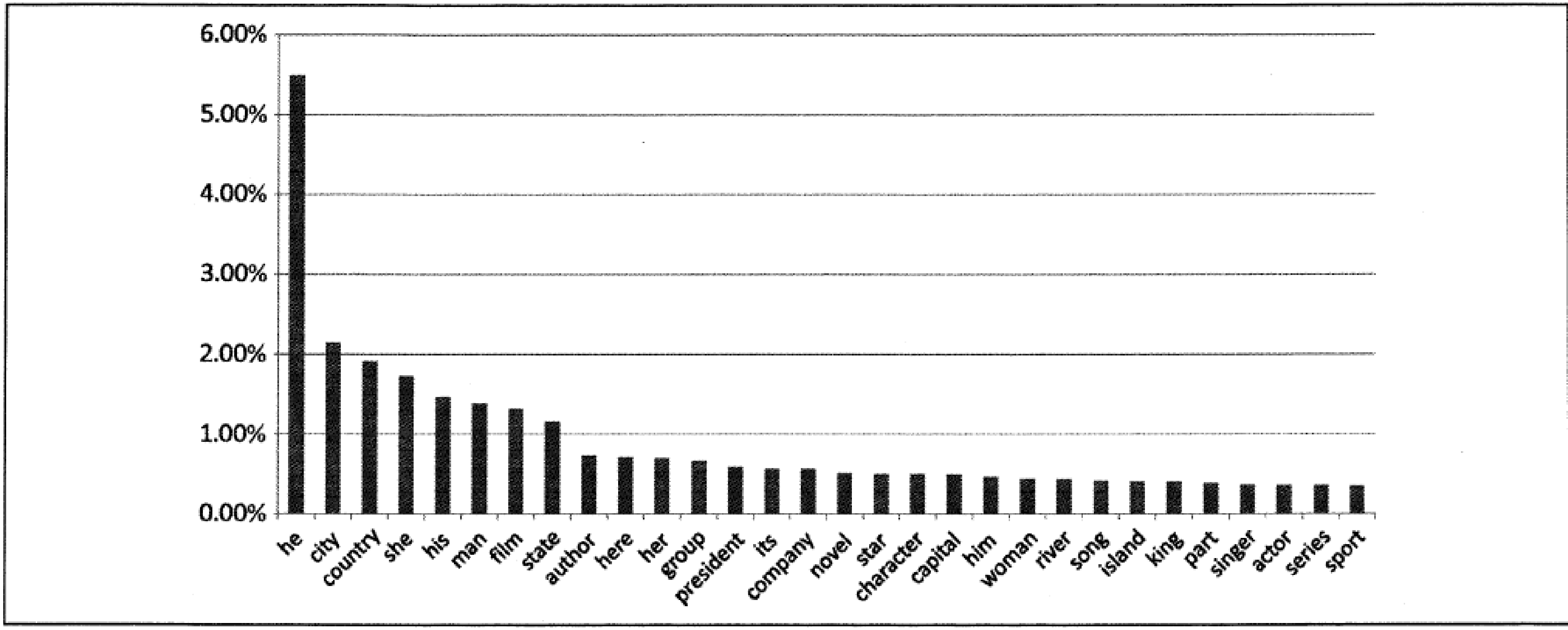


Figure 1

Distribution of the 30 most frequent lexical answer types in 20,000 Jeopardy! questions.

Source: IBM Journal of R&D 56(3-4), 2012

Question Analysis

Train more classifiers

QClass	*Description*	*Example questions (correct answer)*	*Frequency (%)*
DEFINITION	A question that contains a definition of the answer	CONSTRUCTION: It can be the slope of a roof, or the gunk used to waterproof it. (Answer: "pitch") CONSTRUCTION: The name of this large beam that supports the joists literally means "something that encircles". (Answer: "a girder")	14.2
CATEGORY-RELATION	The answer has a semantic relation to the question, where the relation is specified in the category	FORMER STATE GOVERNORS: Nelson A. Rockefeller. (Answer: "New York") COUNTRIES BY NEWSPAPER: Haaretz, Yedioth Ahronoth. (Answer: "Israel")	7.2
FITB	A fill-in-the-blank question asks for completion of a phrase	COMPLETE IT: Attributed to Lincoln: "The ___ is stronger than the bullet." (Answer: "ballot") SHAKESPEARE IN LOVE: "Not that I loved Caesar less," says Brutus, "but that I loved" this city "more." (Answer: "Rome")	3.8
ABBREVIATION	The answer is an expansion of an abbreviation in the question	MILITARY MATTERS: Abbreviated SAS, this elite British military unit is similar to the USA's Delta Force. (Answer: "the Special Air Service")	2.9
PUZZLE	A puzzle question: the answer requires derivation, synthesis, inference, etc.	BEFORE & AFTER: 13th Century Venetian traveler who's a Ralph Lauren short sleeve top with a collar. (Answer: "Marco Polo shirt") THE HIGHEST-SCORING SCRABBLE WORD: Zoom, quiz or heaven. (Answer: "quiz")	2.3
ETYMOLOGY	A question asking for an English word derived from a foreign word having a given meaning	ARE YOU A FOOD"E"?: From the Spanish for "to bake in pastry", it's South America's equivalent of a calzone. (Answer: "an empanada")	1.9
VERB	Question asks for a verb	THE NOT-SO-DEADLY SINS: To capitalize all text in an email is an abomination that signifies the person is doing this. (Answer: "shouting")	1.5
TRANSLATION	A question asking for translation of a word or phrase from one language to another	FRUITS IN FRENCH: Pomme. (Answer: "apple")	1.1
NUMBER	The answer is a number	YOU NEED TO CONVERT: One eighth of a circle equals this many degrees. (Answer: "45")	1.0
BOND	The question asks for what is in common between a set of entities	EDIBLE COMMON BONDS: Mung, snap, string. (Answer: "bean")	0.7
MULTIPLE-CHOICE	The question contains multiple possible answers from which to choose the correct answer	THE SOUTHERNMOST CAPITAL CITY: Helsinki, Moscow, Bucharest. (Answer: "Bucharest") OSCAR, GRAMMY OR BOTH: Mickey Rooney. (Answer: "Oscar")	0.5
DATE	A question asking for a date or year	THE TEENS: World War I ended in November of this year. (Answer: "1918")	0.3

Source:
IBM Journal of R&D 56(3-4), 2012

IBM Watson: Deep QA Architecture

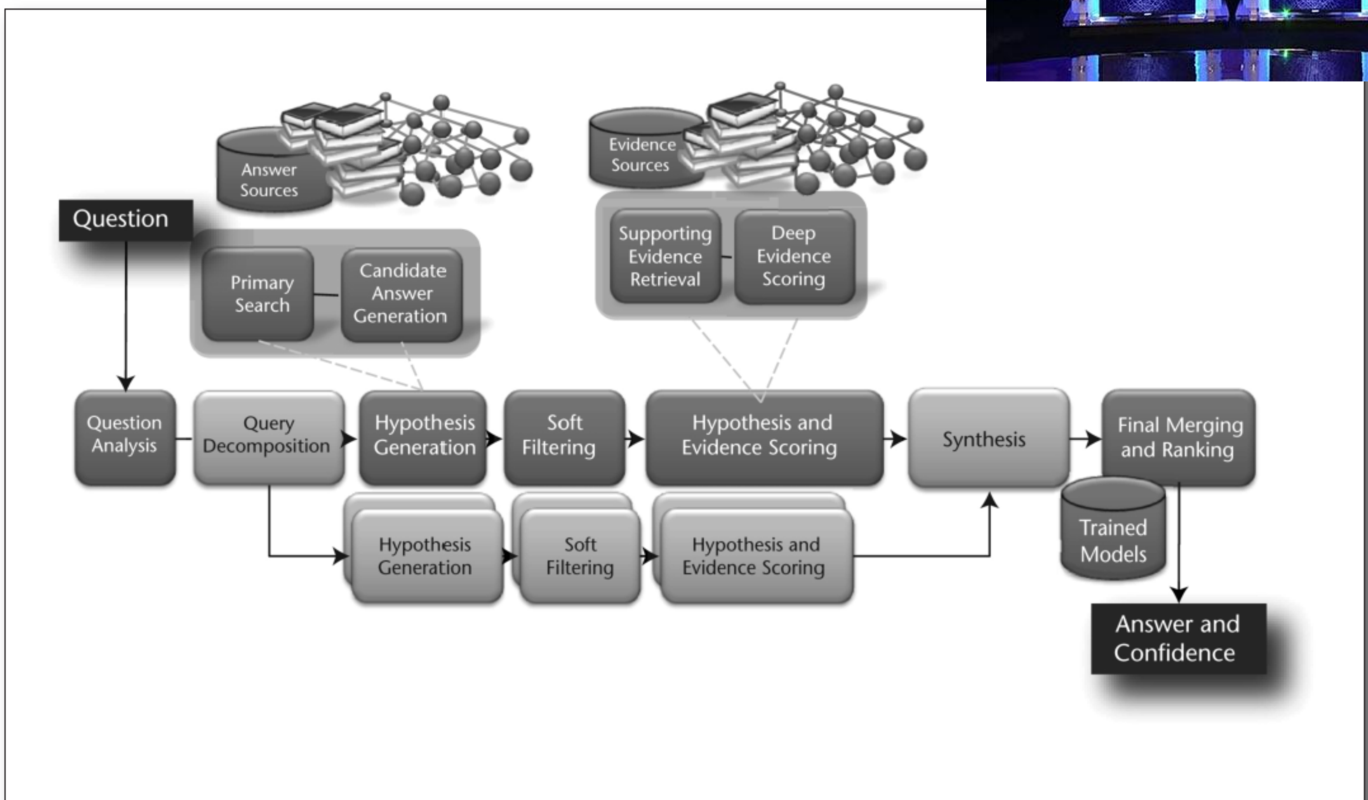


Figure 6. DeepQA High-Level Architecture.

Source: D. Ferrucci et al.: Building Watson. AI Magazine, Fall 2010.

IBM Watson: Deep QA Architecture

question

Overall architecture of Watson (simplified)

Question Analysis: Classification Decomposition → Hypotheses Generation (Search): Answer Candidates → Hypotheses & Evidence Scoring → Candidate Filtering & Ranking

answer

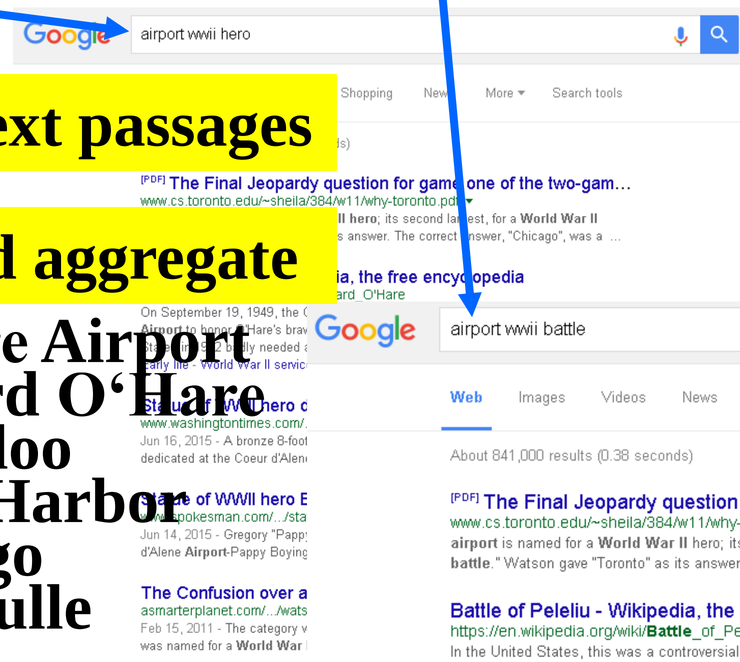
[IBM Journal of R&D 56(3-4), 2012]

IBM Watson: From Question to Answers

(IBM Watson 14-16 Feb 2011)

decompose question

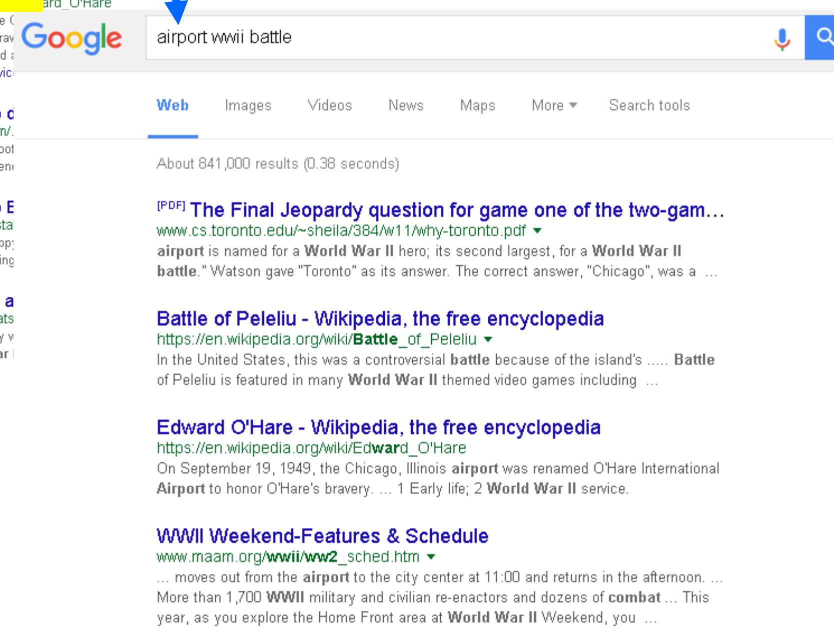
This US city has two airports named for a World War II hero and a World War II battle



find text passages

extract names and aggregate

O'Hare Airport
Edward O'Hare
Waterloo
Pearl Harbor
Chicago
De Gaulle
Paris
New York
……
……



check semantic types

Scoring of Semantic Answer Types

Check for 1) Yago classes, 2) Dbpedia classes, 3) Wikipedia lists

Match lexical answer type against class candidates
based on string similarity and class sizes (popularity)
Examples: Scottish inventor → inventor, star → movie star

Compute **scores for semantic types**, considering:
class match, subclass match, superclass match,
sibling class match, lowest common ancestor, class disjointness, …

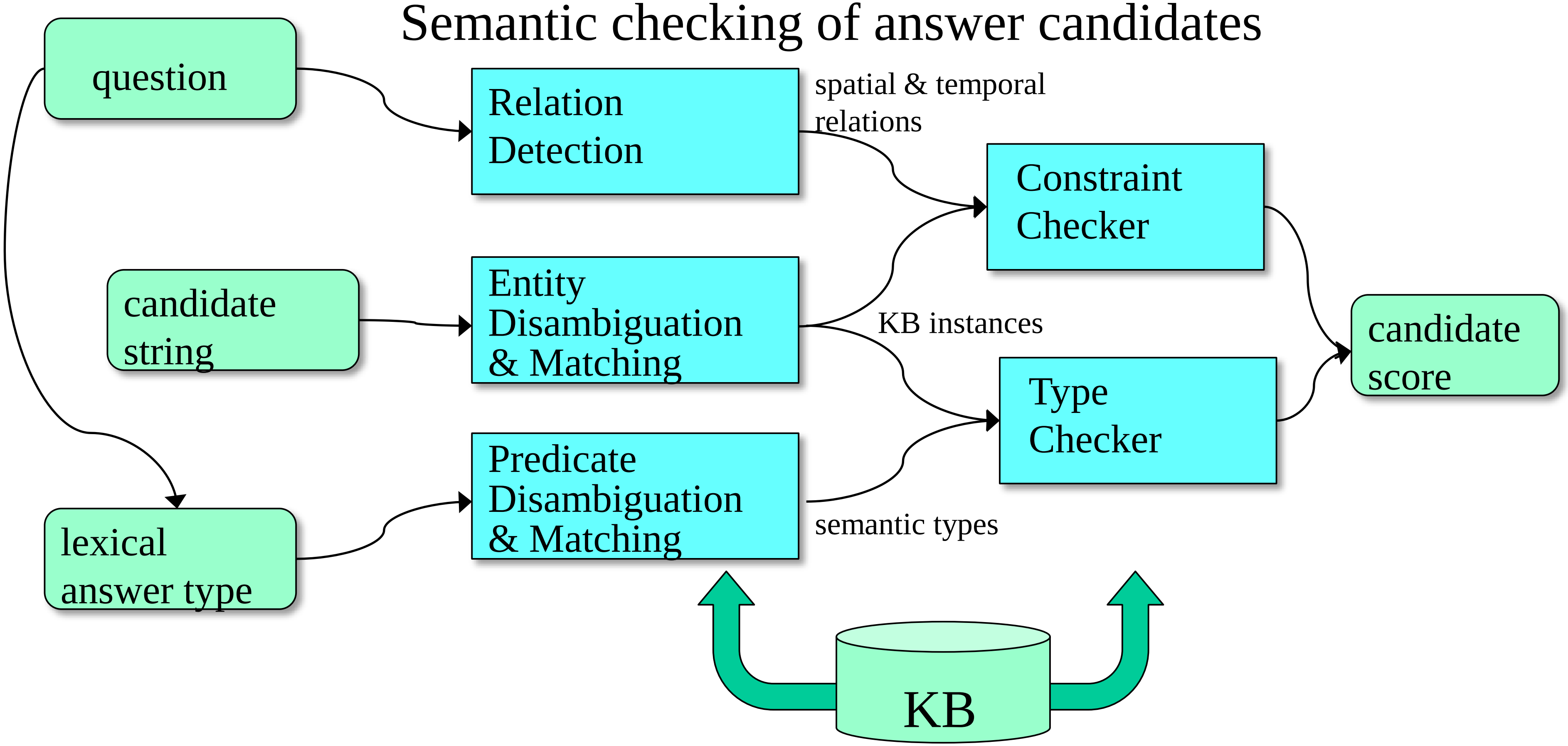
	no types	Yago	Dbpedia	Wikipedia	all 3
Standard QA accuracy	50.1%	54.4%	54.7%	53.8%	56.5%
Watson accuracy	65.6%	68.6%	67.1%	67.4%	69.0%

[A. Kalyanpur et al.: ISWC 2011]

Semantic Technologies in IBM Watson

[A. Kalyanpur et al.: ISWC 2011]

Semantic checking of answer candidates



QA with Structured Data & Knowledge

This town is known as "Sin City" & its downtown is "Glitter Gulch"

Q: **Sin City ?**

  → **movie, graphical novel, nickname for city, …**

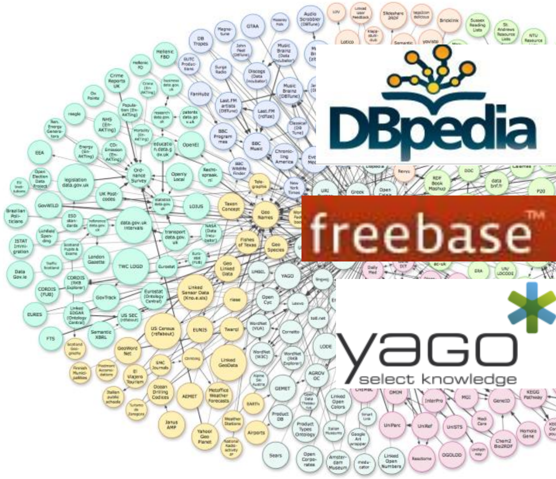
A: **Vegas ? Strip ?**

  → **Vega (star), Suzanne Vega, Vincent Vega, Las Vegas, …**

  → **comic strip, striptease, Las Vegas Strip, …**

question → **structured query**

```
Select ?t Where {
  ?t type location .
  ?t hasLabel "Sin City" .
  ?t hasPart ?d .
  ?d hasLabel "Glitter Gulch" . }
```



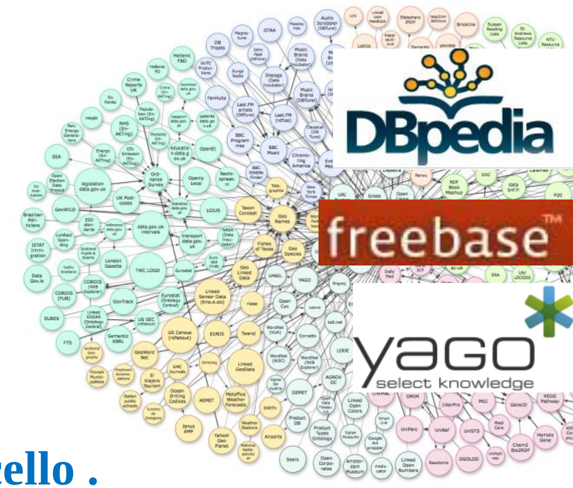
Linked Data
Big Data
Web tables

QA with Structured Data & Knowledge

Which classical cello player covered a
composition from The Good, the Bad, the Ugly?

Q: Good, Bad, Ugly ?
covered ?
A: western movie ? Big Data – NSA - Snowden ?
played ? performed ?

question ? **structured query**



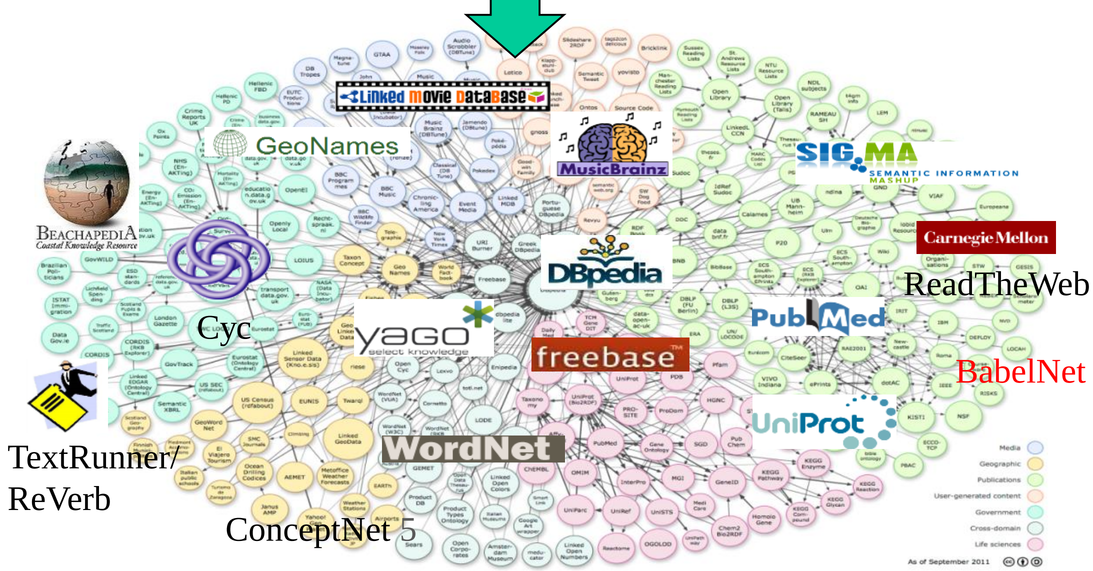
Select ?m Where {
?m type musician . ?m playsInstrument cello .
?m performed ?c . ?c partOf ?f .
?f type movie .
? hasLabel "The Good, the Bad, the Ugly". }

Linked Data
Big Data
Web tables

QA on Web of Data & Knowledge

Who composed scores for westerns and is from Rome?

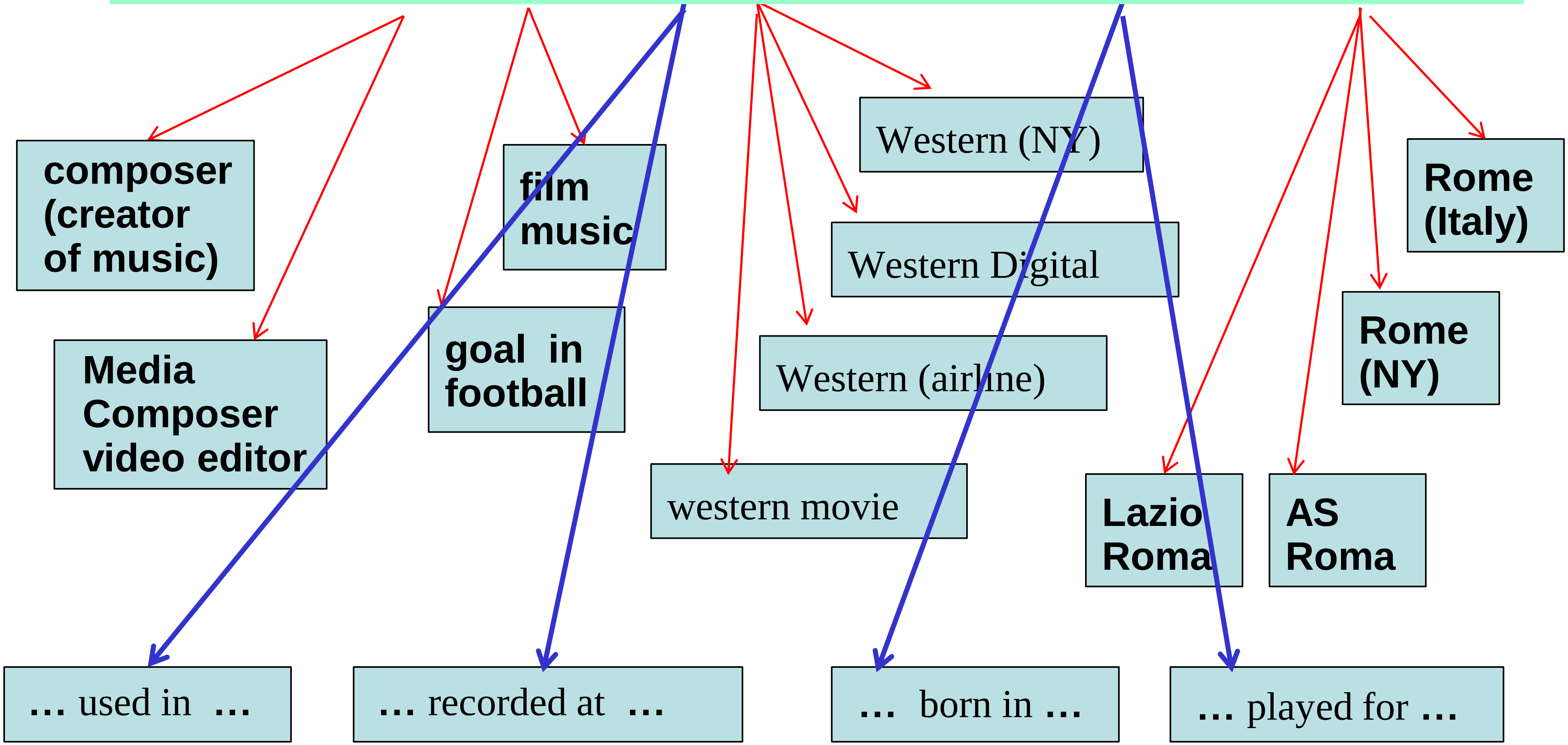
Select ?x Where { ?x created ?s .
?s contributesTo ?m .
?m type westernMovie .
?x bornIn Rome . }



Linked Data
Big Data
Web tables

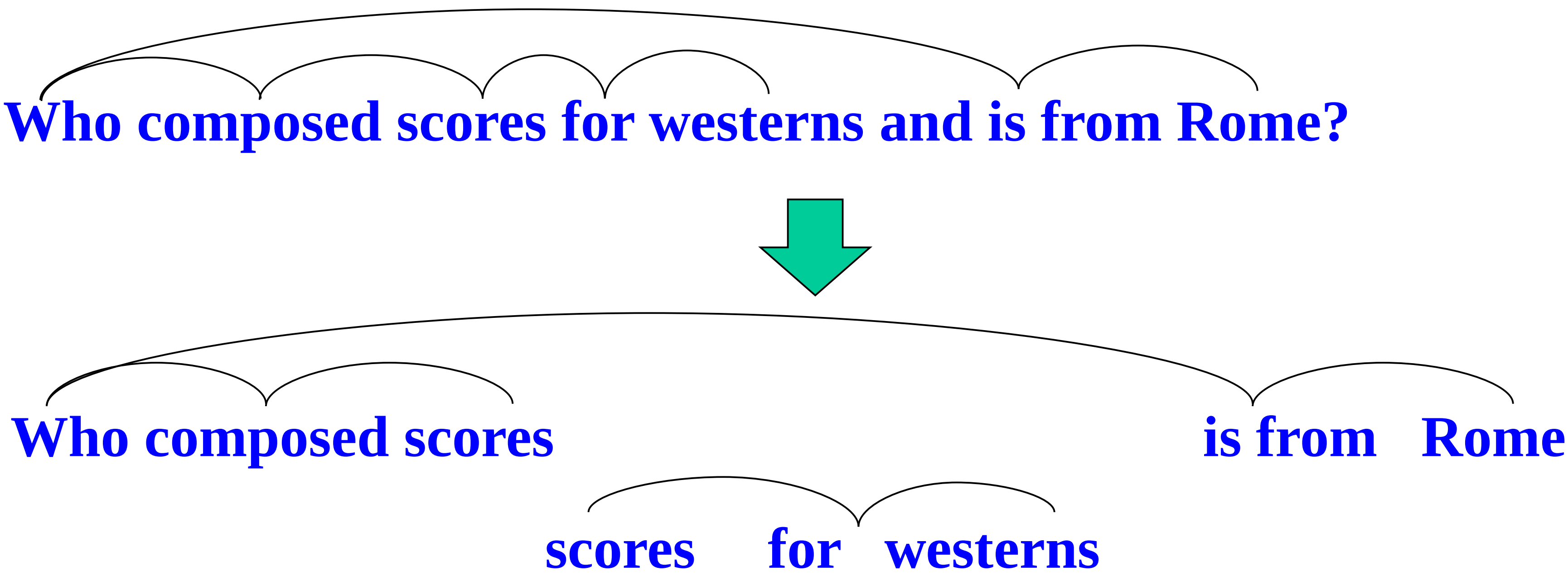
Ambiguity of Relational Phrases

Who composed scores for westerns and is from Rome?



From Questions to Queries

- dependency parsing to decompose question
- mapping of phrases onto entities, classes, relations
- generating SPO triploids (later triple patterns)



Semantic Parsing: from Triploids to SPO Triple Patterns

Map names into entities or classes, phrases into relations

Who composed scores → ?x created ?s
?x type composer
?s type music

scores for westerns → ?s contributesTo ?y
?y type westernMovie

Who is from Rome → ?x bornIn Rome

Paraphrases of Relations

composed (<musician>, <song>) **covered (<musician>, <song>)**

Dylan wrote his song Knockin‘ on Heaven‘s Door, a cover song by the Dead
Morricone ‘s masterpiece is the Ecstasy of Gold, covered by Yo-Yo Ma
Amy‘s souly interpretation of Cupid, a classic piece of Sam Cooke
Nina Simone‘s singing of Don‘t Explain revived Holiday‘s old song
Cat Power‘s voice is sad in her version of Don‘t Explain
Cale performed Hallelujah written by L. Cohen

covered by: (Amy,Cupid), (Ma, Ecstasy), (Nina, Don‘t), (Cat, Don‘t), (Cale, Hallelujah), …

voice in version of: (Amy,Cupid), (Sam, Cupid), (Nina, Don (Cat, Don‘t), (Cale, Hallelujah), …

performed: (Amy,Cupid), (Amy, Black), (Nina, Don (Col, Hallelujah), (Dylan, Knockin), …

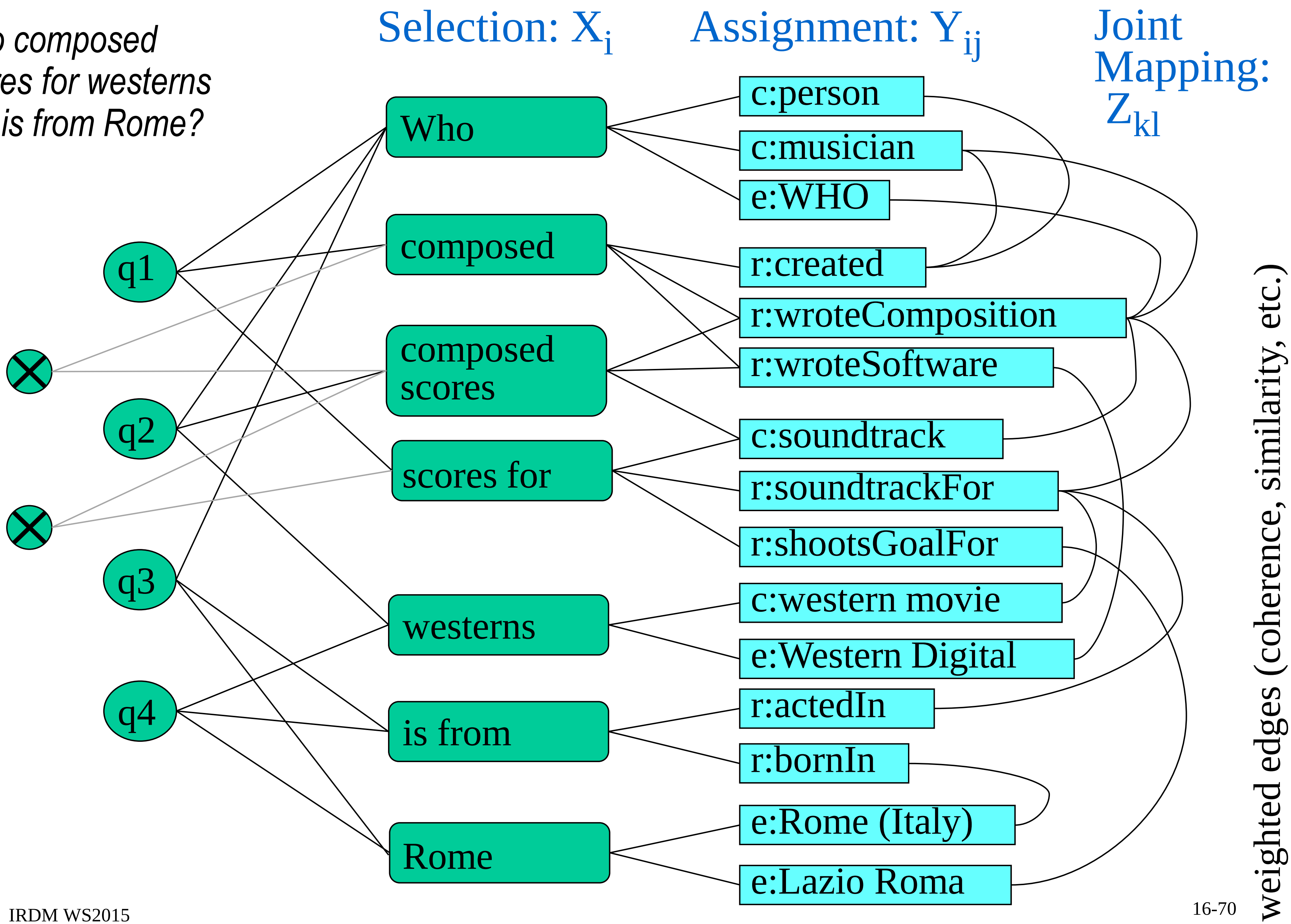
Sequence mining and statistical analysis yield equivalence classes of relational paraphrases

covered (<musician>, <song>):
cover song, interpretation of, singing of, voice in … version , …

composed (<musician>, <song>):
wrote song, classic piece of, ‘s old song, written by, composition of, …

Disambiguation Mapping for Semantic Parsing

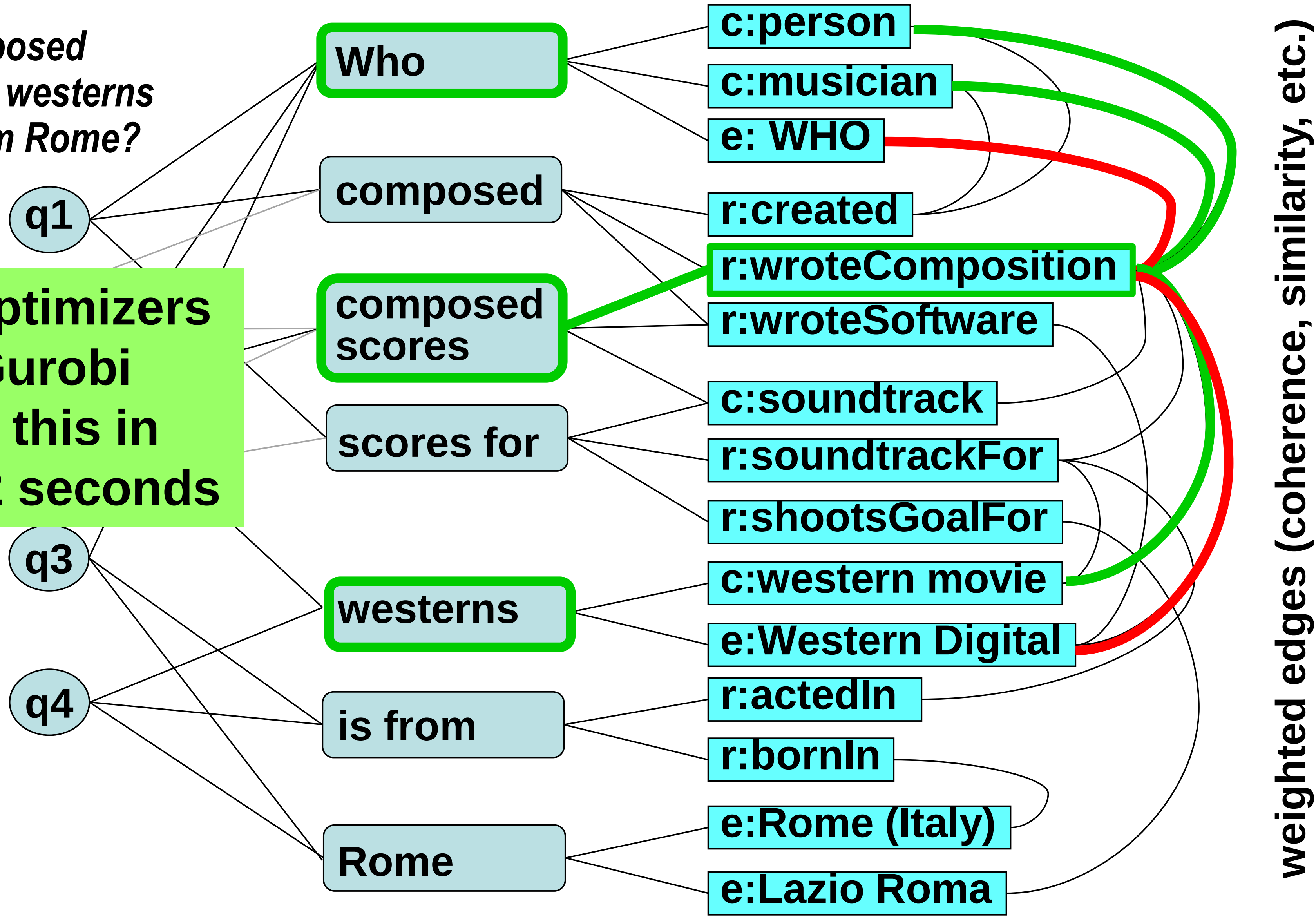
Who composed scores for westerns and is from Rome?



Disambiguation Mapping

[M.Yahya et al.: EMNLP'12, CIKM'13]

Who composed scores for westerns and is from Rome?



ILP optimizers like Gurobi solve this in 1 or 2 seconds

Combinatorial Optimization by ILP (with type constraints etc.)

Prototype for Question-to-Query-based QA

DEANNA

Which composer wrote scores for films and was awarded the Oscar? Submit

Show Sample Questions · Show Advanced Options

Structured Query

```
?x created ?y .
?x type wordnet_composer_109947232 .
?y type wordnet_movie_106613686 .
?x hasWonPrize Academy_Award
```

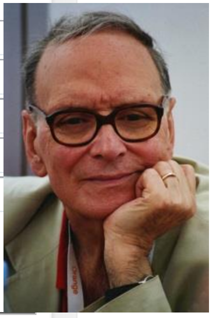
Try it out

YAGO 2 spotlx

Query

Id	Subject	Property	Object	Time		Location	
?id0:	?x	created	?y				
?id1:	?x	type	wordnet_composer_10				
?id2:	?y	type	wordnet_movie_10661:				
?id3:	?x	hasWonPrize	Academy_Award				
?id4:							

query



Results

Summary of Chapter 16

- **Entity search** and ER search over text+KG or text+DB can boost the expressiveness and precision of search engines
- Ranking models for entity answers build on LM‘s and PR/HITS
- Entity search crucially relies on prior information extraction with **entity linking** (Named Entity Recognition and Disambiguation)
- Entity linking combines context similarity, prior popularity and joint coherence into graph algorithms
- **Natural language QA** involves question analysis, passage retrieval, candidate pruning (by KG) and answer ranking
- Mapping questions to structured queries requires general sense disambiguation (for entities, classes and relations)

Additional Literature for 16.1

- K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, L. Si: Expertise Retrieval, Foundations and Trends in Information Retrieval 6(2-3), 2012
- K. Balog, M. Bron, M. de Rijke, Query modeling for entity search based on terms, categories, and examples. ACM TOIS 2011
- H. Fang, C. Zhai: Probabilistic Models for Expert Finding. ECIR 2007
- Z. Nie, J.R. Wen, W.-Y. Ma: Object-level Vertical Search. CIDR 2007
- Z. Nie et al.: Web object retrieval. WWW 2007
- J.X. Yu, L. Qin, L. Chang: Keyword Search in Databases, Morgan & Claypool 2009
- V. Hristidis et al.: Authority-based keyword search in databases. ACM TODS 2008
- G. Kasneci et al.: NAGA: Searching and Ranking Knowledge, ICDE 2008
- H. Bast et al.: ESTER: efficient search on text, entities, and relations. SIGIR 2007
- H. Bast, B. Buchhold: An index for efficient semantic full-text search. CIKM 2013
- H. Bast et al.: Semantic full-text search with broccoli. SIGIR 2014:
- J. Hoffart et al.: STICS: searching with strings, things, and cats. SIGIR 2014
- S. Elbassuoni et al.: Language-model-based ranking for queries on RDF-graphs. CIKM 2009:
- S. Elbassuoni, R. Blanco: Keyword search over RDF graphs. CIKM 2011:
- X. Li, C. Li, C.Yu: Entity-Relationship Queries over Wikipedia. ACM TIST 2012
- M. Yahya et al.: Relationship Queries on Extended Knowledge Graphs, WSDM 2016

Additional Literature for 16.2

- J.R. Finkel: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. ACL 2005
- V. Spitkovsky et al.: Cross-Lingual Dictionary for EnglishWikipedia Concepts. LREC 2012
- W. Shen, J. Wang, J. Han: Entity Linking with a Knowledge Base, TKDE 2015
- Lazic et al.: Plato: a Selective Context Model for Entity Resolution, TACL 2015
- S. Cucerzan: Large-Scale Named Entity Disambiguation based on Wikipedia Data. EMNLP'07
- Silviu Cucerzan: Name entities made obvious. ERD@SIGIR 2014
- D. N. Milne, I.H. Witten: Learning to link with wikipedia. CIKM 2008
- J. Hoffart et al.: Robust Disambiguation of Named Entities in Text. EMNLP 2011
- M.A. Yosef et al.: AIDA: An Online Tool for Accurate Disambiguation of Named Entities in Text and Tables. PVLDB 2011
- J. Hoffart et al.: KORE: keyphrase overlap relatedness for entity disambiguation. CIKM'12
- L.A. Ratinov et al.: Local and Global Algorithms for Disambiguation to Wikipedia. ACL 2011
- P. Ferragina, U. Scaiella: TAGME: on-the-fly annotation of short text fragments CIKM 2010
- F. Piccinno, P. Ferragina: From TagME to WAT: a new entity annotator. ERD@SIGIR 2014:
- B. Hachey et al.: Evaluating Entity Linking with Wikipedia. Art. Intelligence 2013

Additional Literature for 16.3

- D. Ravichandran, E.H. Hovy: Learning surface text patterns for a Question Answering System. ACL 2002:
- IBM Journal of Research and Development 56(3), 2012, Special Issue on “This is Watson”
- D.A. Ferrucci et al.: Building Watson: Overview of the DeepQA Project. AI Magazine 2010
- D.A. Ferrucci et al.: Watson: Beyond Jeopardy! Artif. Intell. 2013
- A. Kalyanpur et al.: Leveraging Community-Built Knowledge for Type Coercion in Question Answering. ISWC 2011
- M. Yahya et al.: Natural Language Questions for the Web of Data. EMNLP 2012
- M. Yahya et al.: Robust Question Answering over the Web of Linked Data, CIKM 2013
- H. Bast, E. Haussmann: More Accurate Question Answering on Freebase. CIKM 2015
- S. Shekarpour et al.: Question answering on interlinked data. WWW 2013:
- A. Penas et al.: Overview of the CLEF Question Answering Track 2015. CLEF 2015
- C. Unger et al.: Introduction to Question Answering over Linked Data. Reasoning Web 2014:
- A. Fader, L. Zettlemoyer, O. Etzioni: Open question answering over curated and extracted knowledge bases. KDD 2014
- T. Khot: Exploring Markov Logic Networks for Question Answering. EMNLP 2015
- J. Berant, P. Liang: Semantic Parsing via Paraphrasing. ACL 2014
- J. Berant et al.: Semantic Parsing on Freebase from Question-Answer Pairs. EMNLP 2013