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

<table>
<thead>
<tr>
<th>text input (keywords)</th>
<th>text output (docs, passages)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard IR</td>
<td>Entity Search Keywords in Graphs (16.1.2)</td>
</tr>
<tr>
<td>Entity Search (16.1.1)</td>
<td>Semantic Web Querying (16.1.3)</td>
</tr>
<tr>
<td>struct. input (entities, SPO patterns)</td>
<td>struct. output (entities, facts)</td>
</tr>
</tbody>
</table>
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
Australian Open: Controlling long rallies and keeping errors in check, Angelique Kerber conquers Ml. Williams

First round - Sun Jan 31 08:28:33 CET 2016

Australian Open: Controlling long rallies and keeping errors in check, Angelique Kerber conquers Ml. Williams by Anand Datta Jan 31, 2016 10:56 IST #AngeliqueKerber #AustralianOpen #AustralianOpen2016 #inMyOpinion #MelbournePark #SerenaWilliams #Serena #Tennis #God, they say, is in the details. Angelique Kerber discovered on Saturday that if you carried your conviction to court and kept the faith in yourself, it is possible to scale a new peak, even if it meant getting past a mountain called Serena Williams. Kerber stuck to her game plan with a monk-like focus to overcome Serena and earn the Australian Open title. Serena is no stranger to formidable German opponents. She was all of 17 when she inflicted a three-set defeat on Steffi Graf for the Indian Wells title nearly seventeen summers ago. It was in the lead up to the French Open in 1999, which was to be the 22nd and last grand slam title of the legendary German's immense career. On Saturday, Williams was at the Australian Open, clearly focused on emulating the great German's collection of grand slam trophies. Angelique Kerber stunned Serena Williams in Melbourne. Getty But there was a supposedly innocuous German across the net - the formidable Steffi's fortress of greatness. At 28, she had finally found her way to a grand slam final for the first time in her 14 years on the tour. And she was facing one I know, a woman who had lost just four of 26 major finals. Not. The Sun in Australia was believed to have printed their back pomegranates of the American was brutally efficient on her way to the final - her grand total of just 26 games. On the other hand, Kerber was first round before battling her way back to emerge an unlikely yet the elated German, as she prepared for the biggest match of her career and try to beat Serena, of course, as well. "I must play my best, she said. "I must play my best at the right moment," added Kerber, who spent a few days holidaying with Steffi at the Agasaki in Las Vegas home in March last year. Working briefly with her idol and soaring in Steffi's thoughts have obviously filled Kerber with a new air of positivity. "I think I'm ready for it because I have a lot of experience in the last few years. I beat top players; I am a top player," Kerber said. "I have a grand slam final and Kerber, despite all her confidence, may have lacked a feeling of very nervous against one of the most formidable opponents tennis may have ever known. Surprisingly, like a woman who woke up on the wrong side of the bed, Serena was all over the place in the first set. The world No. 1 made 23 unforced errors in an easy first set, helping Kerber set down inside the imposing environs of the Rod Laver Arena. The German kept her end of the net steady, making just three errors, as she took advantage of her opponent's sloppy start. Williams though is an intimidating opponent - her guttural screams, the intensity in her eyes and the power behind her shots have forced many an opponent into meek submission, from a position of advantage. Angelique Kerber then hastened to say hello to Serena. Victoria Azarenka. When she gathered herself to take the second set, the momentum was firmly back in her corner. After being 12-23 in the first set, Serena had 16 winners to 5 unforced errors in the second set, a much-improved effort. It felt for a moment that the bout may have ended, with Serena set to assert herself. After all, the 34-year-old veteran had never lost the third set of a grand slam final (9-0) and won every final that she reached in Melbourne (6-0). Till Saturday, 30 January 2016, that is. Kerber though seemed to have acquired nerves of steel on her way to the final. She was threading the needle in the second game, passing Serena at will to draw level at 1-1 early in the final set for a 2-0 lead. Serena summoned her willpower to reel Kerber back immediately to draw level at 2-2. Also see

Entails in this article

Serena Angelique Australian Steffi Andre Australia Agnieszka Indian Victoria French

IRDM WS2015
User types names → system needs to map them entities (in real-time)

Task:

given an input prefix $e_1 \ldots e_k x$ 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

$$\alpha \text{ similarity } (x,e) + \beta \text{ popularity}(e) + \gamma \sum_{i=1..k} \text{ relatedness}(e_i,e)$$
Construct language models for queries $q$ and answers $a$

\[
score(a, q) = \lambda P[q | a] + (1 - \lambda) P[q] \quad \sim KL(LM(q) | LM(a))
\]

with smoothing

$q$ is entity, $a$ is doc $\rightarrow$ 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 $\rightarrow$ analogous
Entity Search: Answer Ranking by Link Analysis

**Entity Authority** (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

Wolf Prize

Nobel Prize

Turing Award

Peter Gruenberg

William Vickrey

Albert Einstein

Vinton Cerf

online ads

usedIn

Internet

TCP/IP

usedIn

TU Darmstadt

workedAt

ETH Zurich

degreeFrom

Stanford

workedAt

Google

spinoff

IRDM WS2015 16-12
16.1.2 Entity Search with Keywords in Graph
Entity Search with Keywords in Graph

Entity-Relationship graph with documents per entity
Entity Search with Keywords in Graph

Entity-Relationship graph with DB records per entity
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)  Journals (JId, Title)
CPublications (PId, Title, CId)          JPublications (PId, Title, Vol, No, Year)
Authors (PId, Person)                   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:**

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

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

Semantic Web Data: Schema-free RDF

SPO triples (statements, facts):
(EnnioMorricone, bornIn, Rome)
(Rome, locatedIn, Italy)
(JavierNavarrete, birthPlace, Teruel)
(Teruel, locatedIn, Spain)
(EnnioMorricone, composed, l‘Arena)
(JavierNavarrete, composerOf, aTale)

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

• **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 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 .
Querying the Structured Web

Structure but no schema: SPARQL well suited

**Wildcards** for properties (relaxed joins):

```
Select ?p, ?c Where {
  ?p instanceOf Composer .
```

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

```
Select ?p, ?c Where {
  ?p instanceOf Composer .
  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

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 hasWon ?a . ?a Name AcademyAward .
  ?p composed ?music [classical, orchestra, cantata, opera] . }

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

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)

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

contextual similarity: mention vs. Entity (bag-of-words, language model)

prior popularity of name-entity pairs
Named Entity Disambiguation (NED)

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

Hurricane, about Carter, is on Bob’s Desire. It is played in the film with Washington.
Named Entity Disambiguation (NED)

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

Coherence: (partial) overlap of (statistically weighted) entity-specific keyphrases

- racism protest song boxing champion wrong conviction
- racism victim middleweight boxing nickname Hurricane falsely convicted
- Grammy Award winner protest song writer film music composer civil rights advocate
- Academy Award winner African-American actor Cry for Freedom film Hurricane film
Named Entity Disambiguation (NED)

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

KB provides building blocks:
- name-entity dictionary,
- relationships, types,
- text descriptions, keyphrases,
- statistics for weights

Hurricane, about Carter, is on Bob’s Desire. It is played in the film with Washington.
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)
Collective Learning with Probabilistic Factor Graphs
[Chakrabarti et al.: KDD’09]:

- model $P[m|e]$ by similarity and $P[e_1|e_2]$ by coherence
- consider likelihood of $P[m_1 \ldots m_k | e_1 \ldots e_k]$
- 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 \( \rightarrow \) approximation algorithms
- Alt.: feature engineering for similarity-only method
  
Coherence Graph Algorithm

- 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
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
• 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 $\text{sim}(\text{cxt}(m_i), \text{cxt}(e_p))$
- coherence $\text{coh}(e_p, e_q)$
- similarity $\text{sim}(\text{cxt}(m_i), \text{cxt}(m_j))$

- 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:
  $\alpha_1 \sum_{ip} \text{sim}(\text{cxt}(m_i), \text{cxt}(e_p))X_{ip} + \alpha_2 \sum_{ijpq} \text{coh}(e_p, e_q)X_{ip}X_{jq}$
  $+ \alpha_3 \sum_{ij} \text{sim}(\text{cxt}(m_i), \text{cxt}(m_j))Z_{iq}$

- 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
- \( \text{sim}(m,e) \) 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 \( P[entity \mid name] \)
Mention-Entity Similarity Edges

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

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

**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 $e$

$$score(e \mid m) \sim \sum_{q \in \text{keyphrases}(e) \text{ in context } (m)} score(q) \cdot \text{dist}(\text{cover}(q), m)^{-\alpha}$$
Entity-Entity Coherence Edges

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

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

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

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

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
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
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
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.
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.
NERD at Work

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

Bruno wrote the score for Himalaya.

Input Type: TEXT  Overall runtime: 1812 ms

Bruno [Bruno Coulais] wrote the score for Himalaya [Himalaya (film)].

Entities Type Filters:

Mention Extraction:

Stanford NER  Manual

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

Fast Mode:

Enabled

Types tag cloud  Focused Types tag cloud
NERD on Tables

Disambiguation Method:
- prior
- prior+sim
- prior+sim+coherence

Parameters: (default should be OK)

Prior-Similarity-Coherence balancing ratio:
- prior VS. sim. balance = 0.4
- (prior+sim.) VS. coh. balance = 0.6

Ambiguity degree = 5

Mention Extraction:
- Stanford NER
- Manual

You can manually tag the mentions by putting them in the list if they are automatically disambiguated in the manual mode.

Input Type: TABLE Overall runtime: 2m, 34s, 101ms
- Types list
- Types tag cloud

Focused Types tag cloud

[Steve Jobs]Steve
[Apple Inc.]Mac
[Dennis Ritchie]Dennis
[C]C
[Richard Stallman]Richard
[GNU Core Utilities]GNU

IRDM WS2015
General Word Sense Disambiguation (WSD)

Which songwriters covered ballads written by the Stones?
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
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)
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?
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
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

---

IBM Journal of R&D 56(3/4), 2012: This is Watson.
More Jeopardy! Questions

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


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

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

IBM Watson: Deep QA Architecture

Overall architecture of Watson (simplified)

question

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)

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

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

...
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

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

<table>
<thead>
<tr>
<th></th>
<th>no types</th>
<th>Yago</th>
<th>Dbpedia</th>
<th>Wikipedia</th>
<th>all 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard QA accuracy</strong></td>
<td>50.1%</td>
<td>54.4%</td>
<td>54.7%</td>
<td>53.8%</td>
<td>56.5%</td>
</tr>
<tr>
<td><strong>Watson accuracy</strong></td>
<td>65.6%</td>
<td>68.6%</td>
<td>67.1%</td>
<td>67.4%</td>
<td>69.0%</td>
</tr>
</tbody>
</table>

[A. Kalyanpur et al.: ISWC 2011]
Semantic Technologies in IBM Watson

[A. Kalyanpur et al.: ISWC 2011]

Semantic checking of answer candidates

- Question
- Candidate string
- Lexical answer type
- Relation Detection
  - Spatial & temporal relations
- Entity Disambiguation & Matching
  - KB instances
- Predicate Disambiguation & Matching
  - Semantic types
- Constraint Checker
- Type Checker
  - KB instances
  - Candidate score
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, …

```
Select ?t Where {
  ?t type location .
  ?t hasLabel "Sin City“ .
  ?t hasPart ?d .
  ?d hasLabel "Glitter Gulch“ . }
```
Which classical cello player covered a composition from The Good, the Bad, the Ugly?

Q: Good, Bad, Ugly? covered?

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“.

QA with Structured 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 .
}
Ambiguity of Relational Phrases

Who composed scores for westerns and is from Rome?

- composer (creator of music)
- film music
- goal in football
- Western (NY)
- Western Digital
- Western (airline)
- Lazio Roma
- AS Roma

- … used in …
- … recorded at …
- … born in …
- … played for …
From Questions to Queries

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

Who composed scores for westerns and is from Rome?
 Semantic Parsing: from Triploids to SPO Triple Patterns

Map names into entities or classes, phrases into relations

Who composed scores \(\rightarrow\) ?x created ?s
?x type composer
?s type music

scores for westerns \(\rightarrow\) ?s contributesTo ?y
?y type westernMovie

Who is from Rome \(\rightarrow\) ?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‘t),
(Cat, Don‘t), (Cale, Hallelujah), …

performed:  (Amy,Cupid), (Amy, Black), (Nina, Don‘t),
(Cohen, Hallelujah), (Dylan, Knockin), …

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, …
Who composed scores for westerns and is from Rome?
Who composed scores for westerns and is from Rome? 

ILP optimizers like Gurobi solve this in 1 or 2 seconds
Prototype for Question-to-Query-based QA

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

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

<table>
<thead>
<tr>
<th>Id</th>
<th>Subject</th>
<th>Property</th>
<th>Object</th>
<th>Time</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>?id3: ?x</td>
<td>created</td>
<td></td>
<td>?y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>?id1: ?x</td>
<td>type</td>
<td></td>
<td>wordnet_composer_10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>?id2: ?y</td>
<td>type</td>
<td></td>
<td>wordnet_movie_106613686</td>
<td></td>
<td></td>
</tr>
<tr>
<td>?id3: ?y</td>
<td>hasWonPrize</td>
<td></td>
<td>Academy_Award</td>
<td></td>
<td></td>
</tr>
<tr>
<td>?id4: ?x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

query
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, M. Bron, M. de Rijke, Query modeling for entity search based on terms, categories, and examples. ACM TOIS 2011
- Z. Nie, J.R. Wen, W.-Y. Ma: Object-level Vertical Search. CIDR 2007
- Z. Nie et al.: Web object retrieval. WWW 2007
- 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
• 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:
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”
- 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
- S. Shekarpour et al.: Question answering on interlinked data. WWW 2013:
- 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
- J. Berant, P. Liang: Semantic Parsing via Paraphrasing. ACL 2014
- J. Berant et al.: Semantic Parsing on Freebase from Question-Answer Pairs. EMNLP 2013