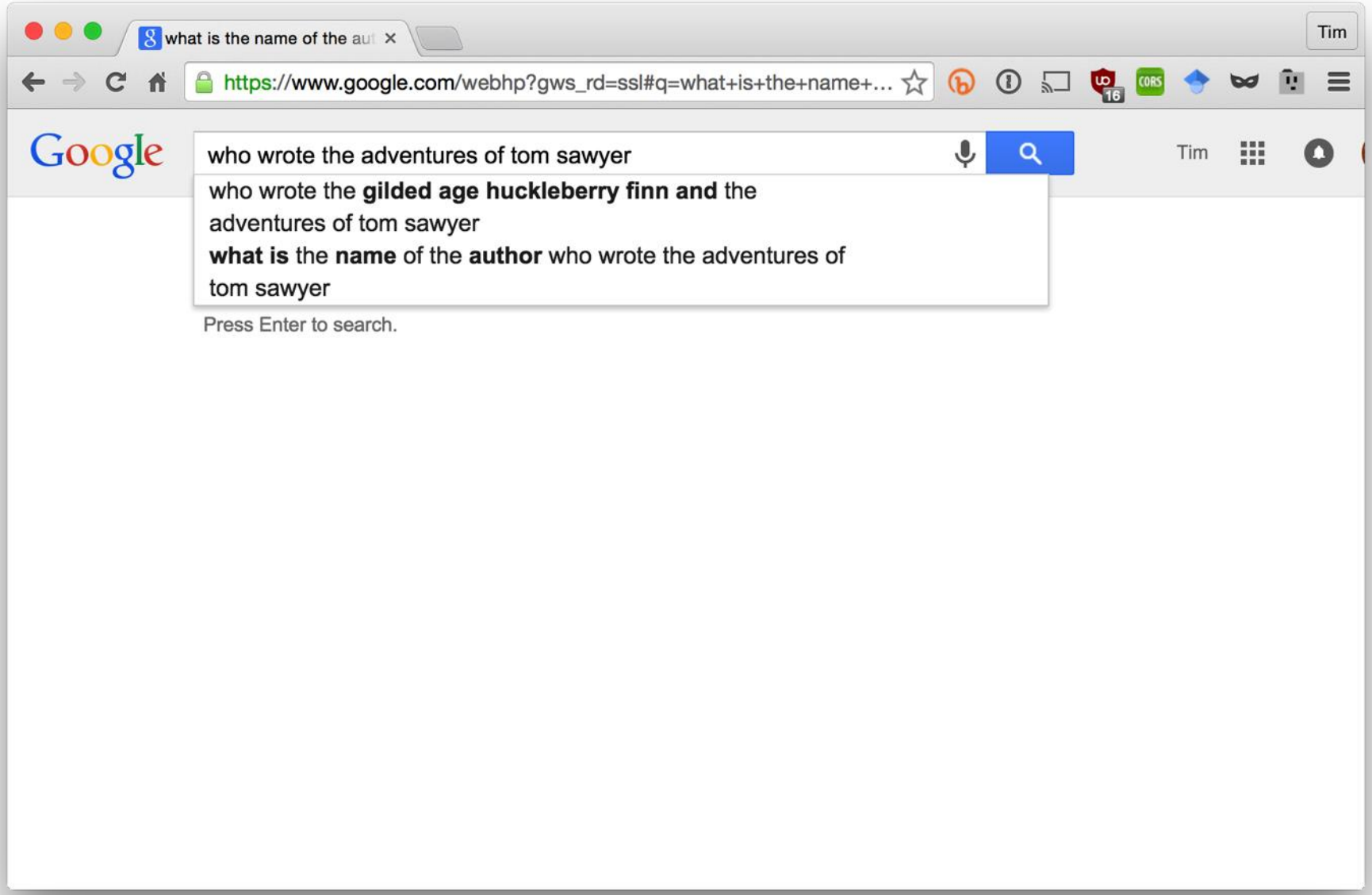
Querying RDF Data with Text Annotated Graphs

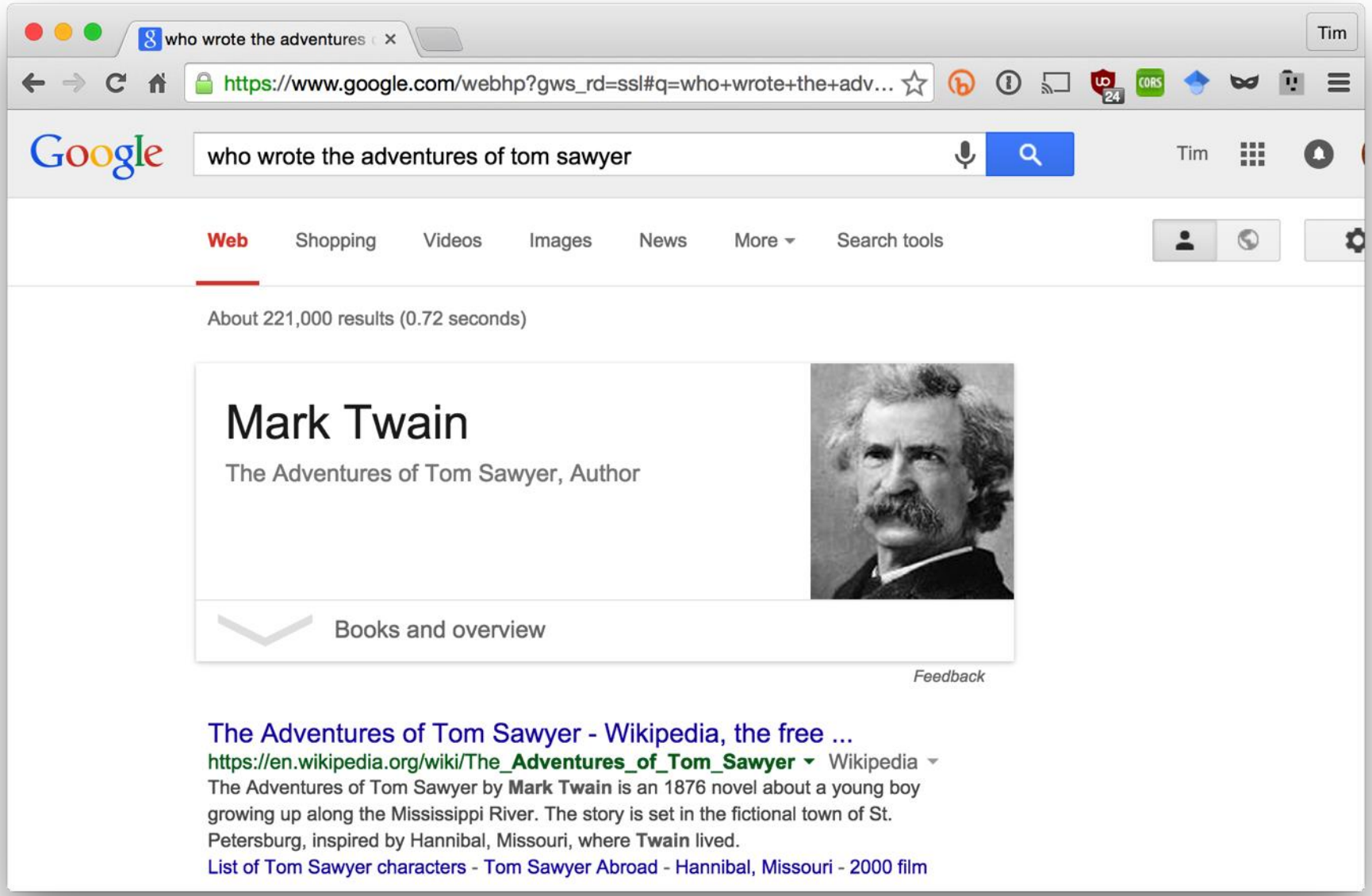
Lushan Han, Tim Finin,
Anupam Joshi and Doreen Cheng

SSDBM'15 • 2015-07-01

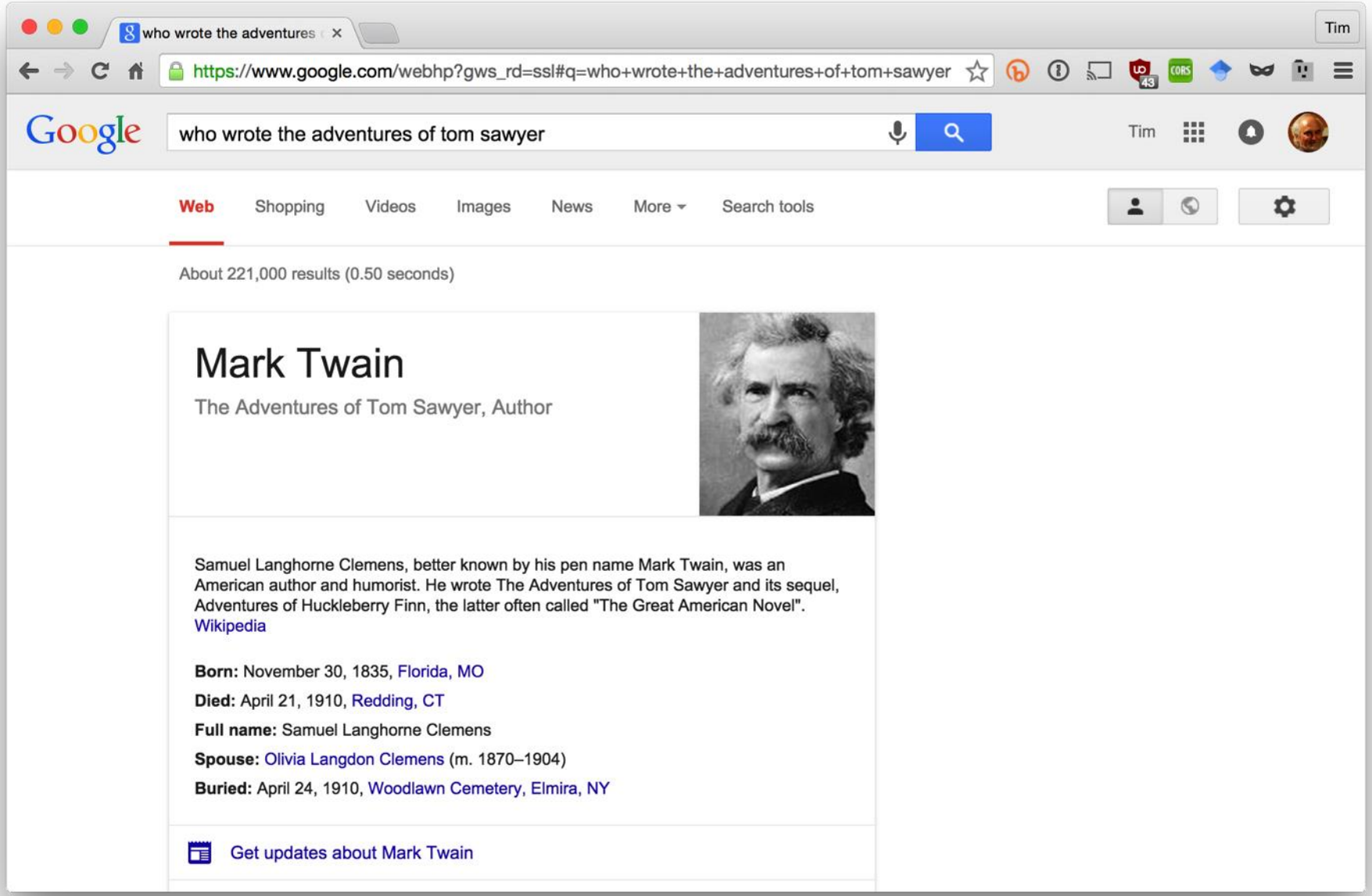
Who wrote Tom Sawyer?



Who wrote Tom Sawyer?



Who wrote Tom Sawyer?



Things, not Strings

Companies moving from *information retrieval* to *questions answering*

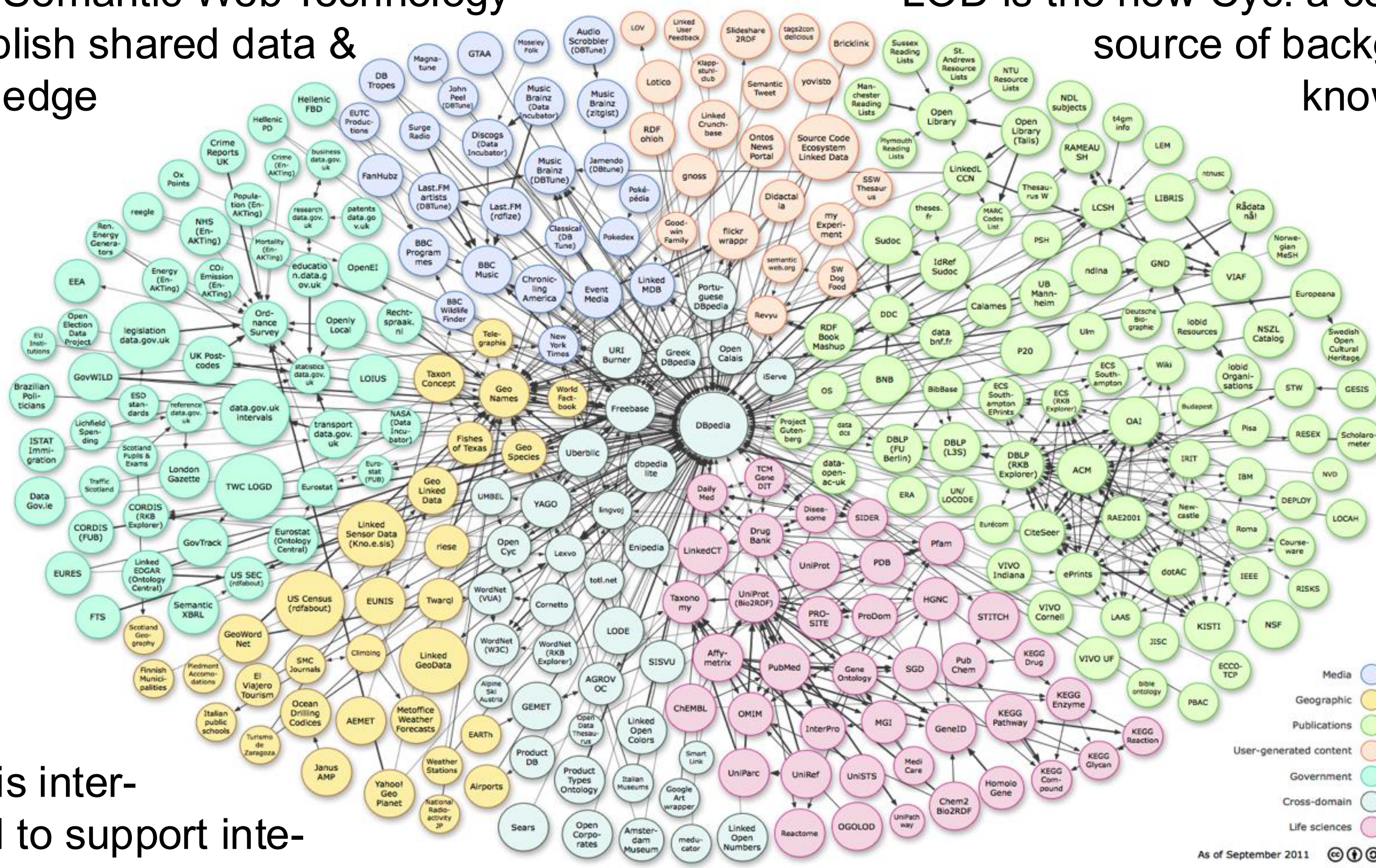
- Need some NLP (e.g. named entity recognition) and good knowledge bases
- Google is a good example
  - 2014: Freebase: 1.2B facts about 43M entities
  - 2015: Google knowledge graph, updated by text IE
- Others have big KBs: Microsoft (Satori), IBM (Watson), Apple, Wolfram (Alpha), Facebook (Open Graph)

DBpedia is an open source KB in RDF that's based on Wikipedia

Linked Open Data Cloud 2007-15

Uses Semantic Web Technology to publish shared data & knowledge

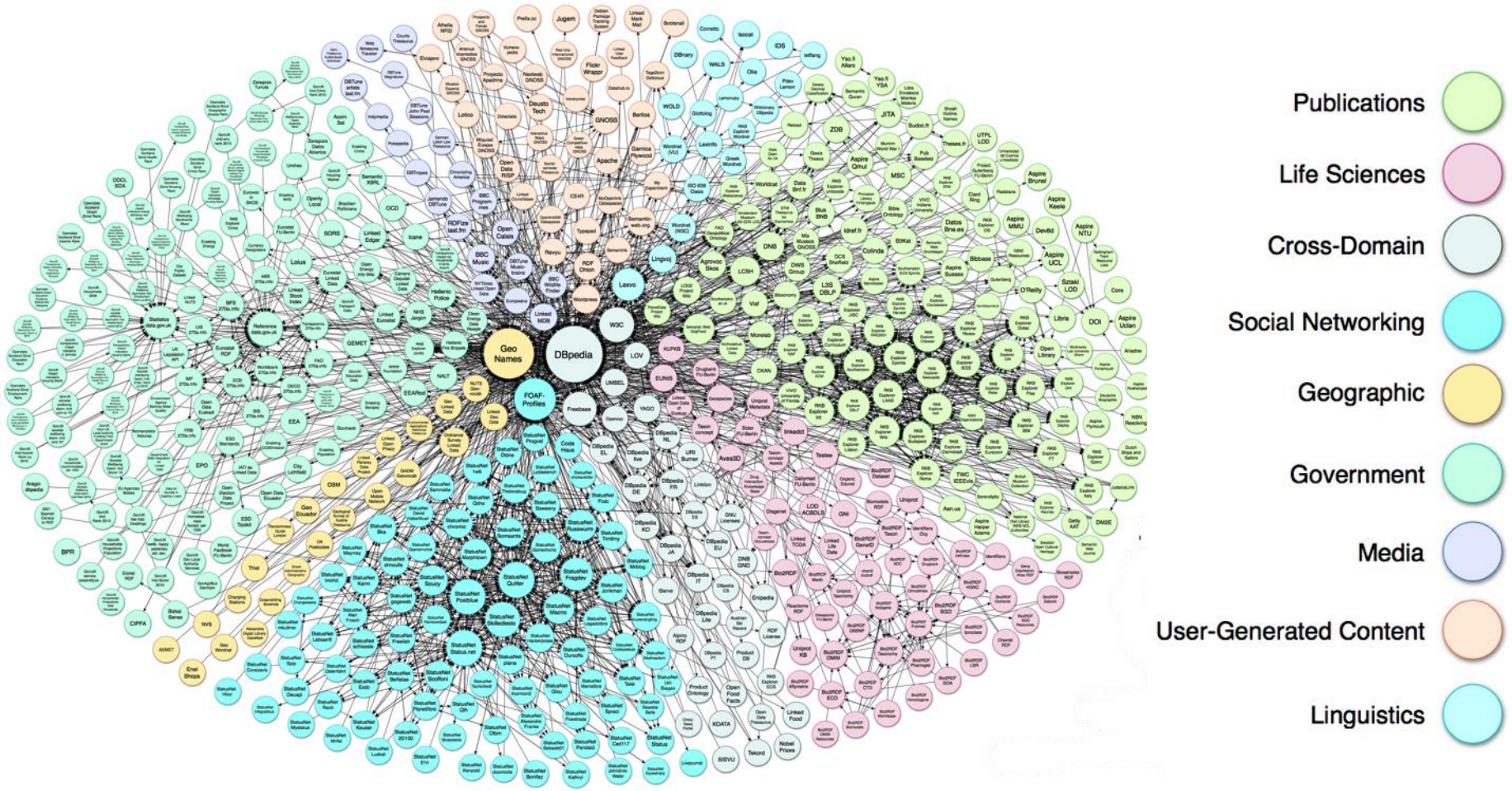
LOD is the new Cyc: a common source of background knowledge



Data is inter-linked to support inte-gration and fusion of knowledge

2011: 31B facts in 295 datasets interlinked by 504M assertions on ckan.net

Linked Open Data Cloud 2015



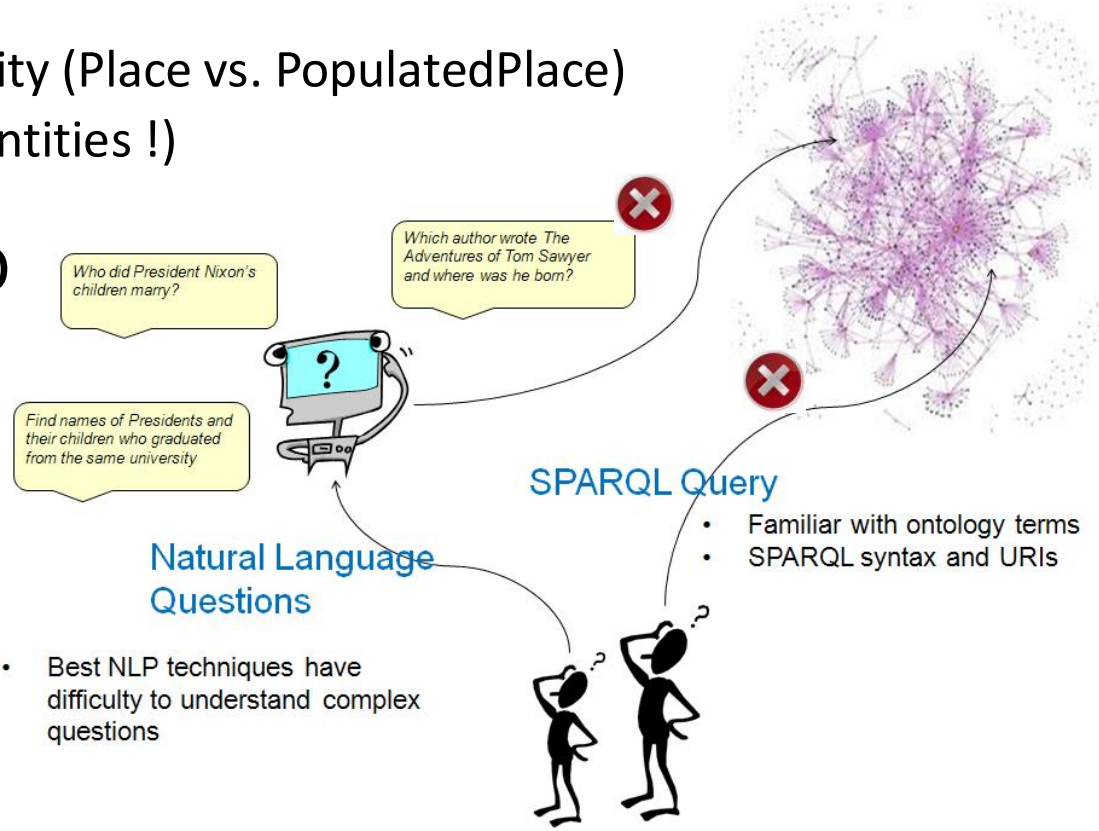
90B facts in 1000 datasets using 3000 vocabularies with 100K classes and 60K properties

DBpedia as Wikipedia LOD

- DBpedia is an important LOD example
  - Structured data from Infoboxes in Wikipedia + raw Wikipedia data
  - RDF in custom ontologies & others, e.g., Yago terms
- Major integration point for LOD cloud
- Datasets for 22 languages
- English DBPedia has:
  - ~400M triples
  - 65M facts (e.g., Alan_Turing born 2012-06-23) about 3.8M entities (person, place, organization…)

LOD is Hard for People to Query

- Querying LOD requires a lot of a user
  - Understand RDF model
  - Master SPARQL, a formal query language
  - Explore large number of ontology terms, hundred classes and thousand properties
  - Deal with term heterogeneity (Place vs. PopulatedPlace)
  - Know relevant URIs (~3M entities !)
- Querying a large LOD is overwhelming
- Natural language query systems still a research goal



Goal

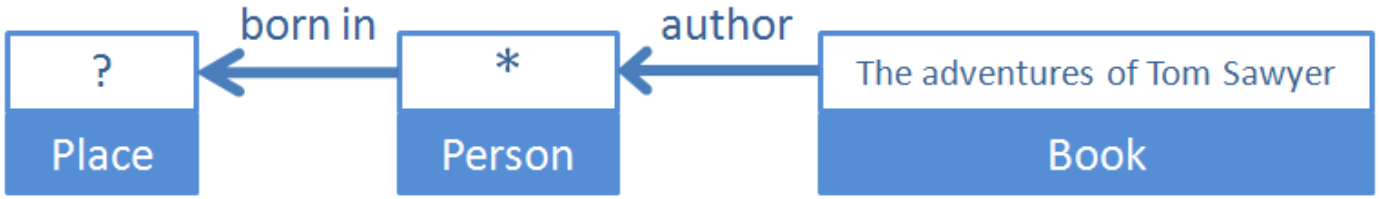
- Develop a system allowing a user with a basic understanding of RDF to query DBpedia and ultimately distributed LOD collections
  - To explore what data is in the system
  - To get answers to questions
  - To create SPARQL queries for reuse or adaptation
- Desiderata
  - Easy to learn and to use
  - Good accuracy, e.g., precision and recall
  - Fast

Key Idea

Reduce problem complexity by having user:

- Enter a simple graph, and
- Annotate the nodes and arcs with words and phrases

Schema-Agnostic Query Interface



Where was the author of the Adventures of Tom Sawyer born?

- Nodes denote entities and links binary relations between them
- Entities described by two freely chosen phrases: its name or value and its concept in the query context
- A *?* marks output entities, a *marks ones to ignore
- Compromise between NLI and SPARQL
  - Users provide compositional structure of question
  - Freedom to use any phrases in annotating structure

Instance data vs. Schema data

- We don’t exploit schema axioms (Actor⊆Person)
- Two key datasets
  - Relation dataset: all relations between instances
  - Type dataset: all type definitions for instances

- Integrate RDF literal data types into five that are familiar to users
  - *ˆNumber, ˆDate, ˆYear, ˆText and ˆLiteral*
  - *ˆLiteral* is the super type of the other four

Automatically enrich set of types

Automatically deduce types from relations

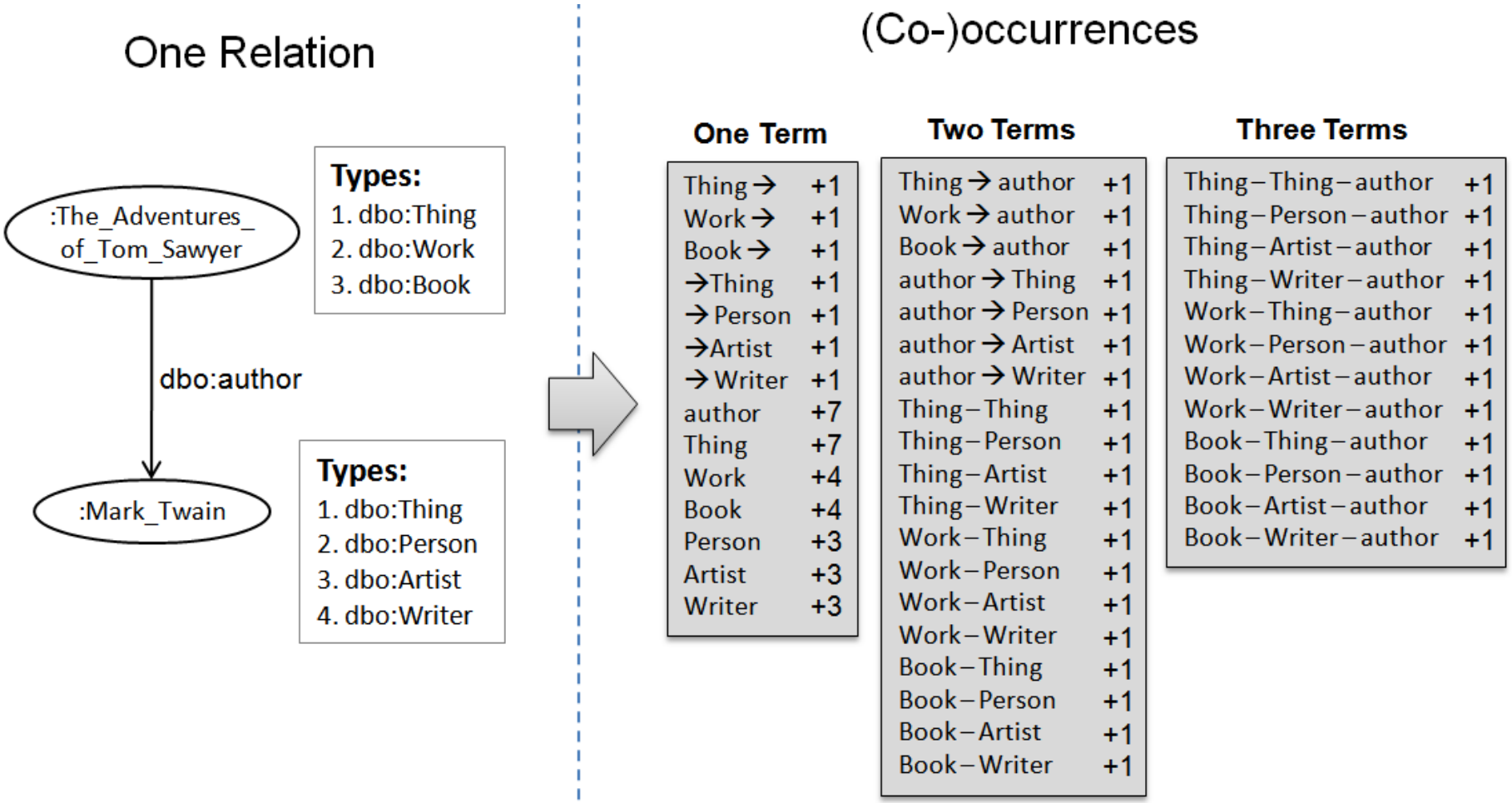
- Infer attribute types from data type properties

  <Beijing>, population, “20693000” => ˆ*Population*

- Infer classes from object properties

  < Zelig>, director, <Woody Allen> => ˜*Director*

Counting Co-occurrence



Concept Association Knowledge

Measured by directed PMI value

1) **Writer→**: @pseudonym 6.0, notableWork 6.0, influencedBy 5.7, skos:subject 5.7, influenced 5.5, movement 5.1, ethnicity 4.3, @birthName 4.3, @deathDate 4.2, relative 4.1, occupation 4.0, @birthDate 3.8, nationality 3.4, education 3.4, child 3.3, award 3.2, deathPlace 3.2, @activeYearsStartYear 3.2, partner 3.2, @activeYearsEndYear 3.1, genre 3.1, spouse 3.0, birthPlace 3.0, citizenship 2.9, foaf:homepage 2.8

2) **→Writer**: author 6.8, influencedBy 6.4, influenced 6.1, basedOn 5.3, illustrator 5.1, writer 5.1, creator 5.1, coverArtist 4.4, executiveProducer 4.4, relative 4.2, translator 4.1, lyrics 4.0, previousEditor 3.9, editor 3.6, spouse 3.5, child 3.4, nobelLaureates 3.3, designer 3.2, partner 3.2, associateEditor 3.2, director 3.0, narrator 3.0, chiefEditor 2.9, storyEditor 2.8, person 2.7

3) **Book→**: @isbn 5.8, @numberOfPages 5.8, @oclc 5.6, mediaType 5.6, @lcc 5.6, literaryGenre 5.6, @dcc 5.5, author 5.4, coverArtist 5.2, @publicationDate 5.1, nonFictionSubject 5.1, illustrator 5.1, translator 4.9, publisher 4.9, series 4.5, language 4.0, subsequentWork 3.3, previousWork 3.2, country 1.7, designer -1.9, @meaning -1.9, @formerCallsign -2.1, @review -2.4, @callsignMeaning -2.5, programmeFormat -2.6

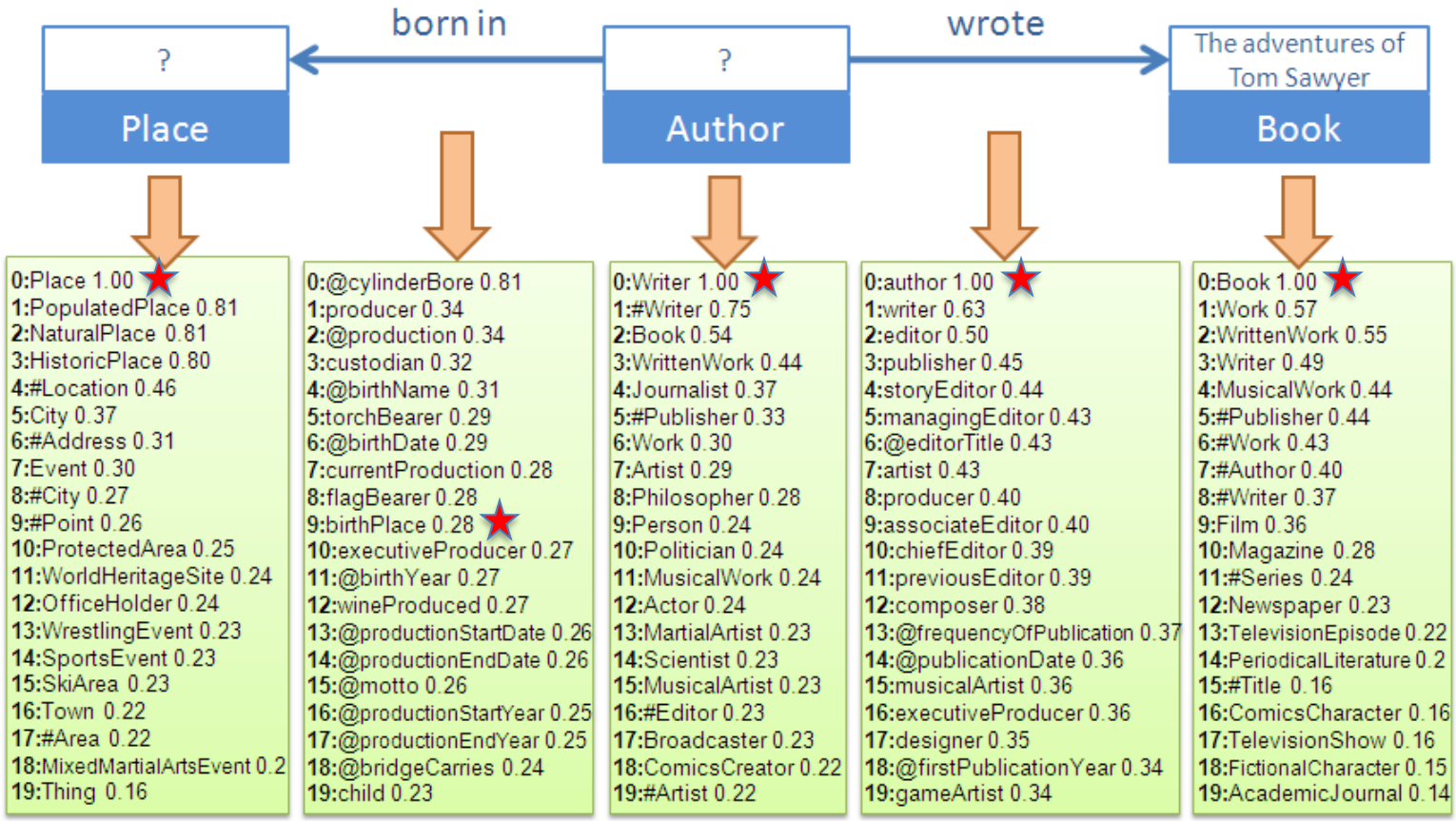
4) **→Book**: notableWork 6.8, firstAppearance 6.4, basedOn 6.1, lastAppearance 5.9, previousWork 5.8, subsequentWork 5.8, series 4.8, knownFor 3.8, notableIdea 3.1, portrayer 2.6, currentProduction 2.3, related 1.9, author 1.7, nonFictionSubject 1.7, writer 1.4, translator 1.1, influencedBy 1.1, significantProject 1.1, award 0.9, coverArtist 0.8, relative 0.5, movement 0.5, associatedMusicalArtist 0.5, associatedBand 0.4, illustrator 0.3

5) **author**: →Writer 6.8, Musical→ 6.1, Play→ 5.4, Book→ 5.4, Website→ 5.4, WrittenWork→ 5.1, →Journalist 5.0, →Philosopher 4.9, →Website 4.8, →Artist 4.5, →Comedian 4.1, →Person 3.9, →ComicsCreator 3.8, →Scientist 3.6, TelevisionShow→ 3.4, Work→ 3.3, →Senator 3.2, →FictionalCharacter 2.8, →PeriodicalLiterature 2.7, →Governor 2.4, →Wrestler 2.3, →MemberOfParliament 2.3, →OfficeHolder 2.3, →Cleric 2.2, →MilitaryPerson 2.2

Translation – Step One

finding semantically similar ontology terms

For each concept or relation in semantic graph, generate *k* most semantically similar candidate ontology classes or properties



Semantic similarity of words

- Words occurring in the same context are similar

  doctor/physician > doctor/nurse > doctor/lawyer > doctor/sandwich

- 3B word corpus http://ebiq.org/r/351

  Stanford WebBase crawl, cleaned, POS tagged, lemmatized

- Vocabulary: 29K terms
  - Content words (noun, verb, adjective, adverb) occurring frequently + some phrases
- ± 4 widow

  is **program_VB** in both **java_NN** and **python_NN** and **run_VB**

SVD to Reduce dimensionality

- LSA Similarity
  - 29k x 29k POS tagged term co-occurrence matrix
  - Replace frequency counts by logs
  - Perform SVD, retaining retain 300 largest singular values
  - Semantic similarity = vector cosine similarity
- Add knowledge from WordNet
  - Address polysemy issue of LSA similarity
  - Boost LSA score using synset, hypernym, derivation and other relations

TOEFL Synonym Evaluation

- Our LSA model is better on some tasks than Google's *word2vec*
- TOEFL synonym task: pick best synonym from a list of four for 80 words

provision

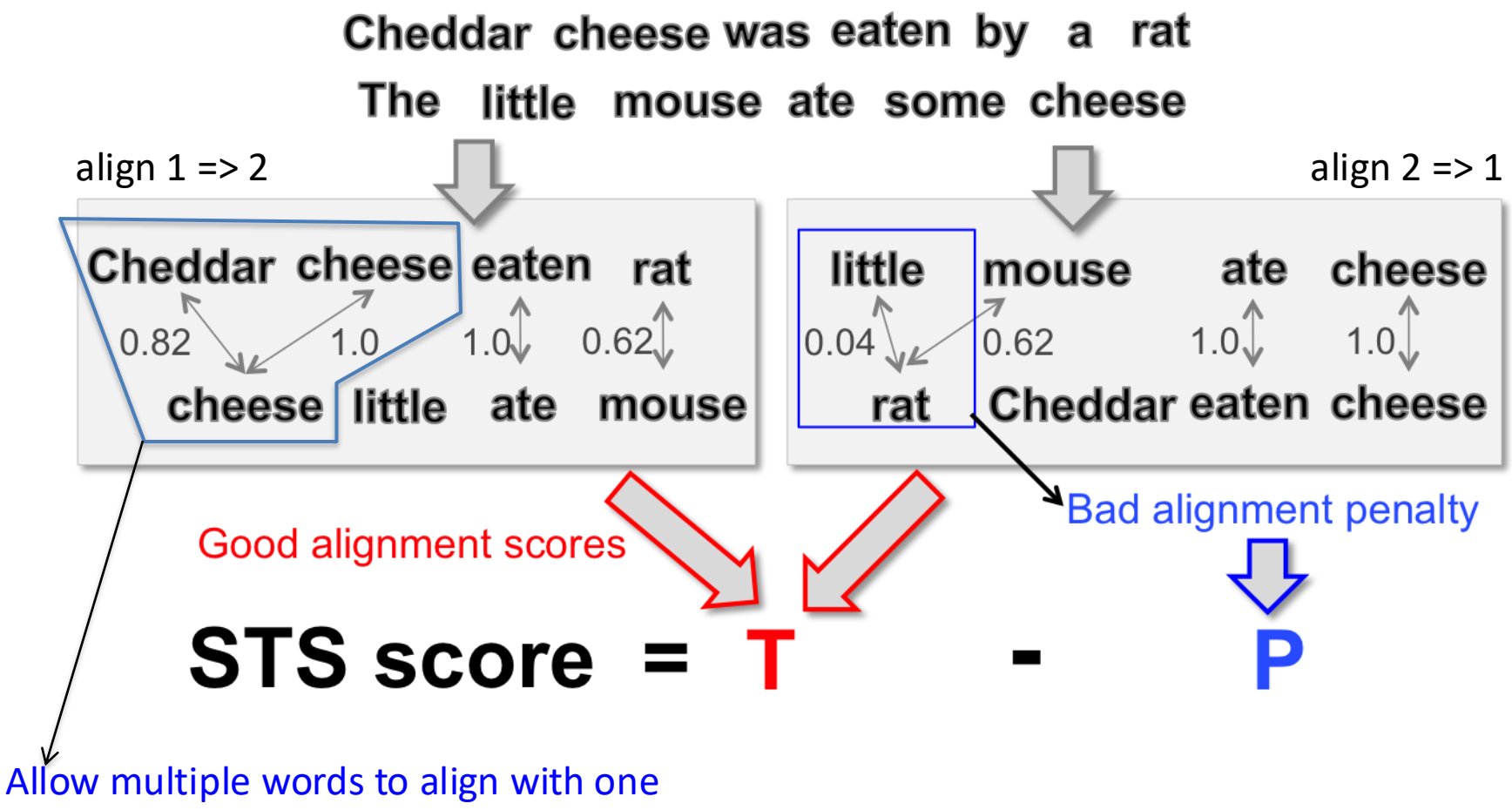
- stipulation
- interrelation
- jurisdiction
- interpretation

	± 1 model	± 4 model	word2vec
Correctly answered	73	76	67
OOV words	halfheartedly	halfheartedly	tranquility
Accuracy	92.4%	96.2%	84.8%

From word to text similarity

- Basic ***align and penalize*** approach
  ① align words to maximize word similarity
  ② compute average word similarity for pairs
  ③ penalize unaligned terms, known antonyms
- Preprocessing: POS tag, lemmatization, REs to identify number and dates, stopword removal
- Word similarity wrapper for numbers, time expressions, pronouns and OOV words
- Penatly component for antonyms, etc.

A&P align and penalize example



Translation – Step Two

disambiguation algorithm

Interpretation goodness metric is degree to which its *ontology terms associate* like corresponding *user terms* connect in SAQ

$$h^* = \underset{h \in H}{argmax}\, \Phi(h, G_q)$$
$$\doteq \underset{h \in H}{argmax} \sum_{i=1}^{m} \Phi(h, L_i)$$

Joint disambiguation

$$A = \overrightarrow{\mathrm{PMI}}(h(s_i), h(r_i)) + \overrightarrow{\mathrm{PMI}}(h(r_i), h(o_i))$$
$$A' = \overrightarrow{\mathrm{PMI}}(h(o_i), h(r_i)) + \overrightarrow{\mathrm{PMI}}(h(r_i), h(s_i))$$
$$(\hat{s}_i,\ \hat{o}_i) = \begin{cases} (o_i,\ s_i), & \text{if } A' - A > \alpha \\ (s_i,\ o_i), & \text{if } A' - A \leqslant \alpha \end{cases}$$

Resolve direction

$$\Phi(h, L_i) = \overrightarrow{\mathrm{PMI}}(h(\hat{s}_i), h(r_i)) \cdot sim(\hat{s}_i, h(\hat{s}_i)) \cdot sim(r_i, h(r_i))$$
$$+\overrightarrow{\mathrm{PMI}}(h(r_i), h(\hat{o}_i)) \cdot sim(\hat{o}_i, h(\hat{o}_i)) \cdot sim(r_i, h(r_i))$$
$$+2 \cdot \mathrm{PMI}(h(\hat{s}_i), h(\hat{o}_i)) \cdot sim(\hat{s}_i, h(\hat{s}_i)) \cdot sim(\hat{o}_i, h(\hat{o}_i))$$

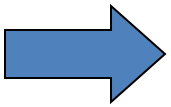
Compute reasonableness for a single link

SPARQL Generation

Translation of semantic graph query to SPARQL is straightforward given mappings

Concepts

- Place => Place
- Author => Writer
- Book => Book



Relations

- born in => birthPlace
- wrote => author

```
PREFIX dbo:<http://dbpedia.org/ontology/>
SELECT DISTINCT ?x, ?y WHERE {
  ?0 a dbo:Book;
       rdfs:label ?label0 .
  ?label0 bif:contains '"Tom Sawyer"' .
  ?x a dbo:Writer .
  ?y a dbo:Place .
  {?0 dbo:author ?x} .
  {?x dbo:birthPlace ?y} .}
```

Evaluation

- 30 test questions from 2011 *Workshop on Question Answering over Linked Data* answerable using DBpedia
- Three human subjects unfamiliar with DBpedia translated the test questions into SAQ queries
- Compare system to FREyA and PowerAqua systems that both require human-crafted domain knowledge. FREyA also requires user to resolve ambiguity.

		30 questions		6 two-relations	
		Prec.	*Recall*	*Prec.*	*Recall*
FREyA		0.829	0.849	0.855	0.789
PowerAqua		0.698	0.757	0.167	0.167
Our system	con., w/ step 3	0.668	0.742	0.780	0.809
	non-empty	0.746	0.816	0.780	0.809

Other Evaluations

- DBLP Database Evaluation
  - Generate SQL to answer questions on DBLP data,
  - Which UCSD authors from have papers in CIKM?
- 2013 and 2014 SemEval tasks
  - Algorithms to measure semantic similarity of short text sequences => top scorer overall
- 2015 SAQ Challenge Evaluation
  - Top system in Schema-Agnostic Query Evaluation Challenge at 2015 Extended Semantic Web Conf.

Conclusion and Future Work

- Baseline system works well for DBpedia and DBLP KBs
- Ongoing and future work
  - Better Web interface
  - Allow user feedback and advice
  - Add entity matching (which George Bush?)
  - Extend and scale to a distributed LOD collection
- For more information, see http://ebiq.org/93