Question Answering (I)
Overview

Instructor: Huan Sun
Computer Science
University of California at Santa Barbara

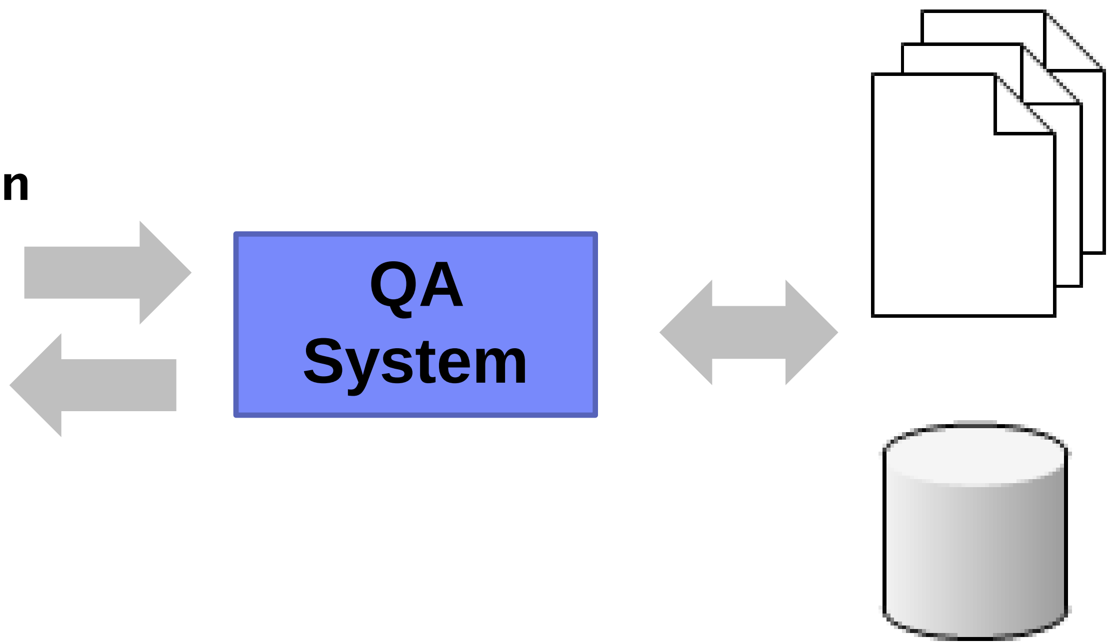
Slides adapted from those by Prof. Dan Jurafsky, Prof. Dan Klein, Dr. André Freitas, Dr. Haklae Kim

What is Question Answering?

Text Corpora

Question: Who is the daughter of Bill Clinton married to?

Answer: Marc Mezvinsky



Database

What is Question Answering?

- A research field on its own.
- Focus on the development and evaluation of approaches and systems to answer questions.
- Multidisciplinary:
  - Natural Language Processing
  - Information Retrieval
  - Knowledge Representation
  - Databases
  - Linguistics
  - Artificial Intelligence
  - Software Engineering
  - ...

Why Question Answering?

- Humans are built-in with natural language communication capabilities.
- Very natural way for humans to communicate information needs.

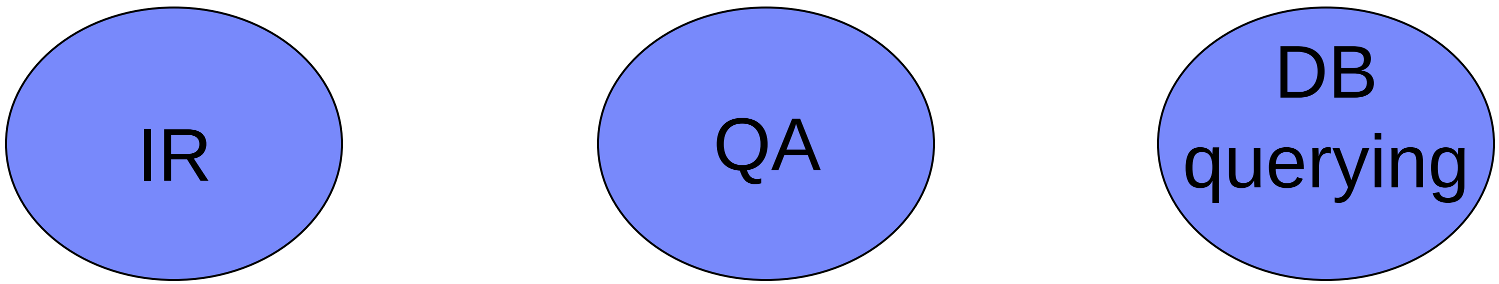
QA in the Big Data Era:

IBM Watson

Facebook Graph search

Google Knowledge Graph

Apple Siri


IR
QA
DB querying

IR

QA

Keyword Search:

1. User still carries the major efforts in interpreting the data.
2. Satisfying information needs may depend on multiple search operations.

Input: Keyword search

Typically specification of simpler information needs.

Output: documents, structured data.

Question Answering:

1. Delegates more ‘interpretation effort’ to the machines.
2. Direct answer.

Input: natural language query

Specification of complex information needs.

Output: direct answer

DB querying

QA

Structured Queries:

1. A priori user effort in understanding the schemas behind databases.
2. Effort in mastering the syntax of a query language.
3. Satisfying information needs may depend on multiple querying operations.

Input: Structured query
Output: data records, aggregations, etc

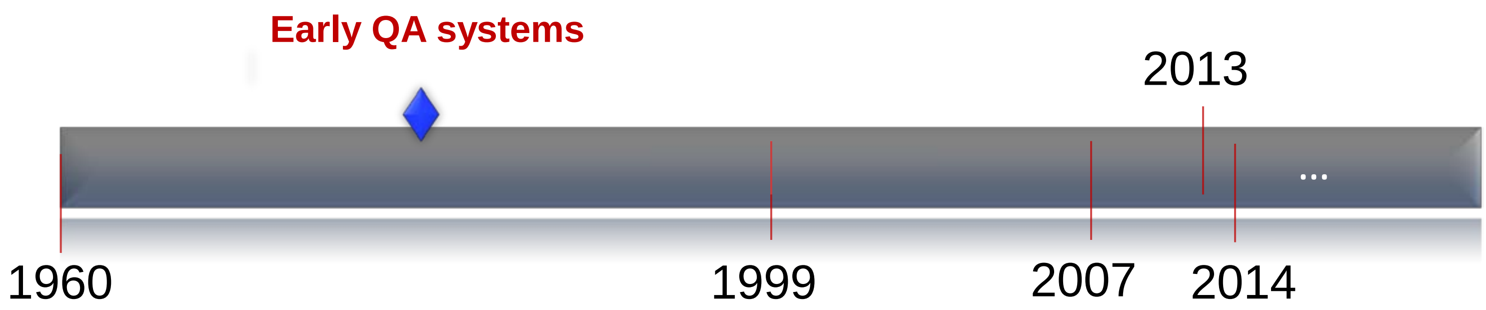
Question Answering:

1. Delegates more ‘interpretation effort’ to the machines.

2. Direct answer.

Input: natural language query
Specification of complex information needs.
Output: direct answer

Question Answering History



- Answering English Questions by Computer: A Survey
  [R. F. Simmons, *Communications of the ACM,*1965]

- Early QA systems (most closed-domain)
  - Natural language database systems
  - Dialog systems
  - Reading comprehension systems

Early QA Systems

Natural Language Database Systems

- Analysing a question to produce a database query, e.g.,

"Give me all lunar samples with Silicon"

⇓

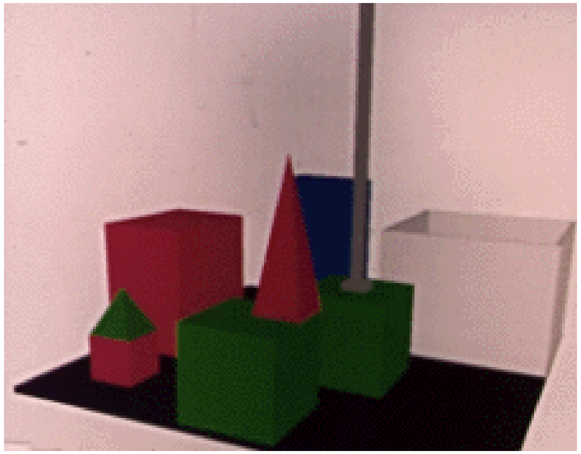
(FOR EVERY X1 / (SEQ SAMPLES) :
(CONTAIN X1 OVERALL SILICON) ; (PRINTOUT X1))

- Example systems such as BASEBALL and LUNAR (see Green et al. 1961 and Woods 1973).

Early QA Systems

Dialog Systems

- To modelling human dialogue.
- Early systems such as SHRDLU were limited to working in a small domain [Winograd, 1972].
- This is still an active research area.



SHRDLU

PERSON: PICK UP A BIG RED BLOCK.
COMPUTER: OK. (does it)

PERSON: GRASP THE PYRAMID.
COMPUTER: I DON'T UNDERSTAND WHICH PYRYMID YOU MEAN.

Early QA Systems

Reading Comprehension Systems

How Maple Syrup is Made

Maple syrup comes from sugar maple trees. At one time, maple syrup was used to make sugar. This is why the tree is called a "sugar" maple tree. Sugar maple trees make sap. Farmers collect the sap. The best time to collect sap is in February and March. The nights must be cold and the days warm. The farmer drills a few small holes in each tree. He puts a spout in each hole. Then he hangs a bucket on the end of each spout. The bucket has a cover to keep rain and snow out. The sap drips into the bucket. About 10 gallons of sap come from each hole.

- Who collects maple sap?
- What does the farmer hang from a spout?
- When is sap collected?
- Where does the maple sap come from?
- Why is the bucket covered?

Early QA Systems

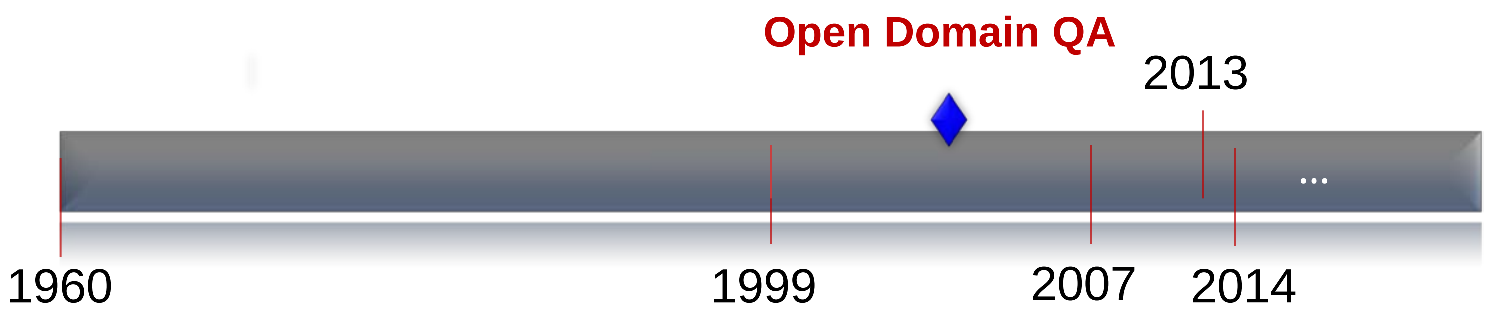
Reading Comprehension Systems

- Systems such as Quarc and Deep Read (see Riloff et al. 2000 and Hirschman et al. 1999) claim results of between 30% and 40%.
- Select the sentence which best answers the question.
- Use a set of pattern matching rules augmented with one or more natural language techniques.

1. Score(S) += WordMatch(Q,S)
2. If $\neg$ contains(Q,NAME) and contains(S,NAME) Then Score(S) += **confident**
3. If $\neg$ contains(Q,NAME) and contains(S,*name*) Then Score(S) += **good_clue**
4. If contains(S,{NAME,HUMAN}) Then Score(S) += **good_clue**

Who Rules
[Quarc, Riloff et al, 2000]

Question Answering History



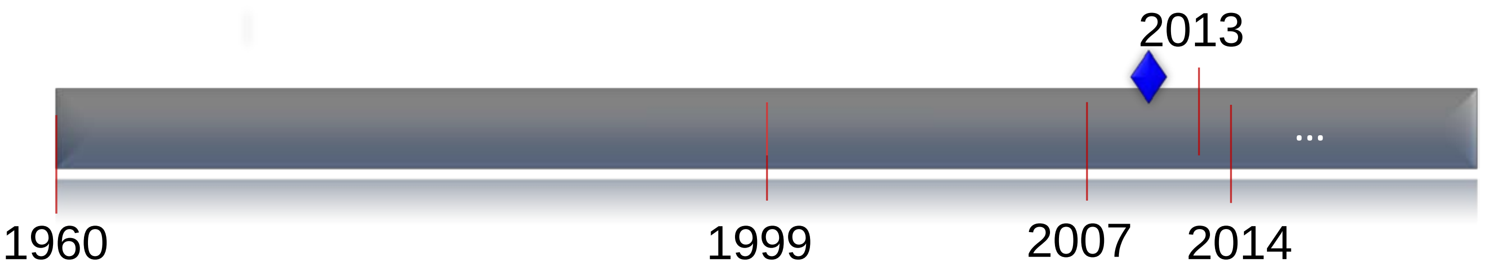
- No restrictions on the scope of questions a user can ask
- TREC Question Answering Track (1999~2007)
- Most systems use large text collections to extract a relevant answer
- The Web as a popular choice of text collection for recent systems

Examples of Open QA systems

- AskJeeves(1996, Ask.com)
- START (1997)
- QuASM (1999)
- IONAUT (2000)
- Mulder (2001)
- LCC (2001)
- Webclopedia (2002)
- AnswerBus (2002)
- AskMSR (2002)
- Ephyra (2006)
- Participants in TREC QA

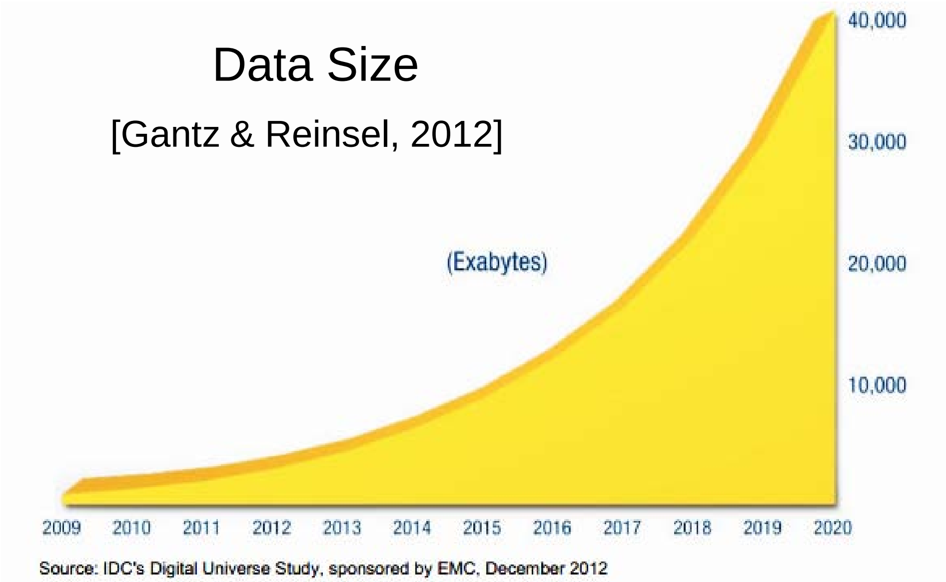
Question Answering History

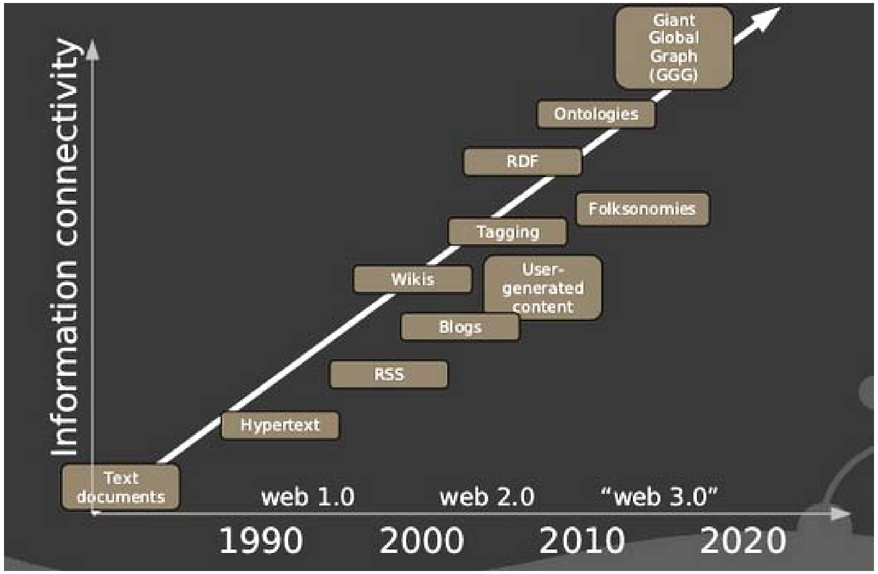
QA over Linked Data



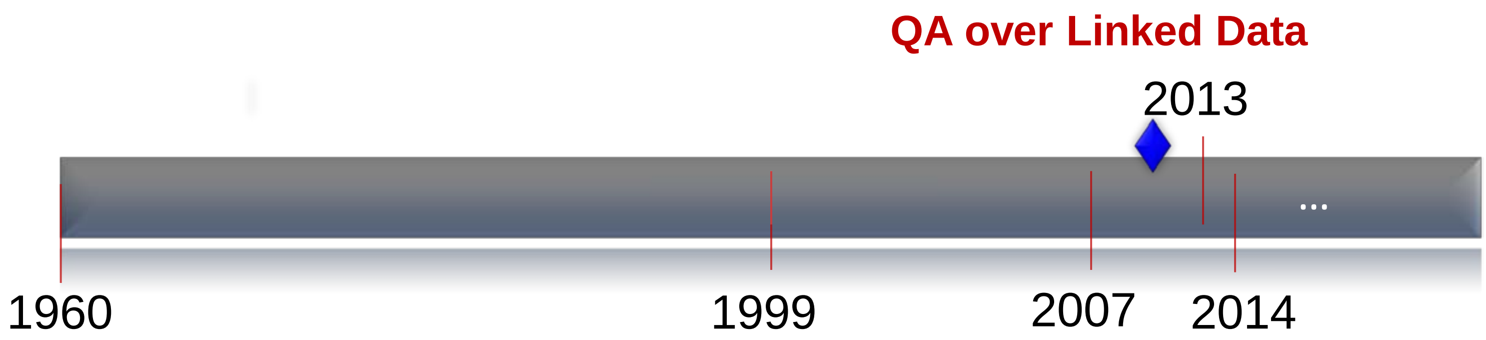
- Data source: Big & linked data

Connectedness [Eifrem, 2009]



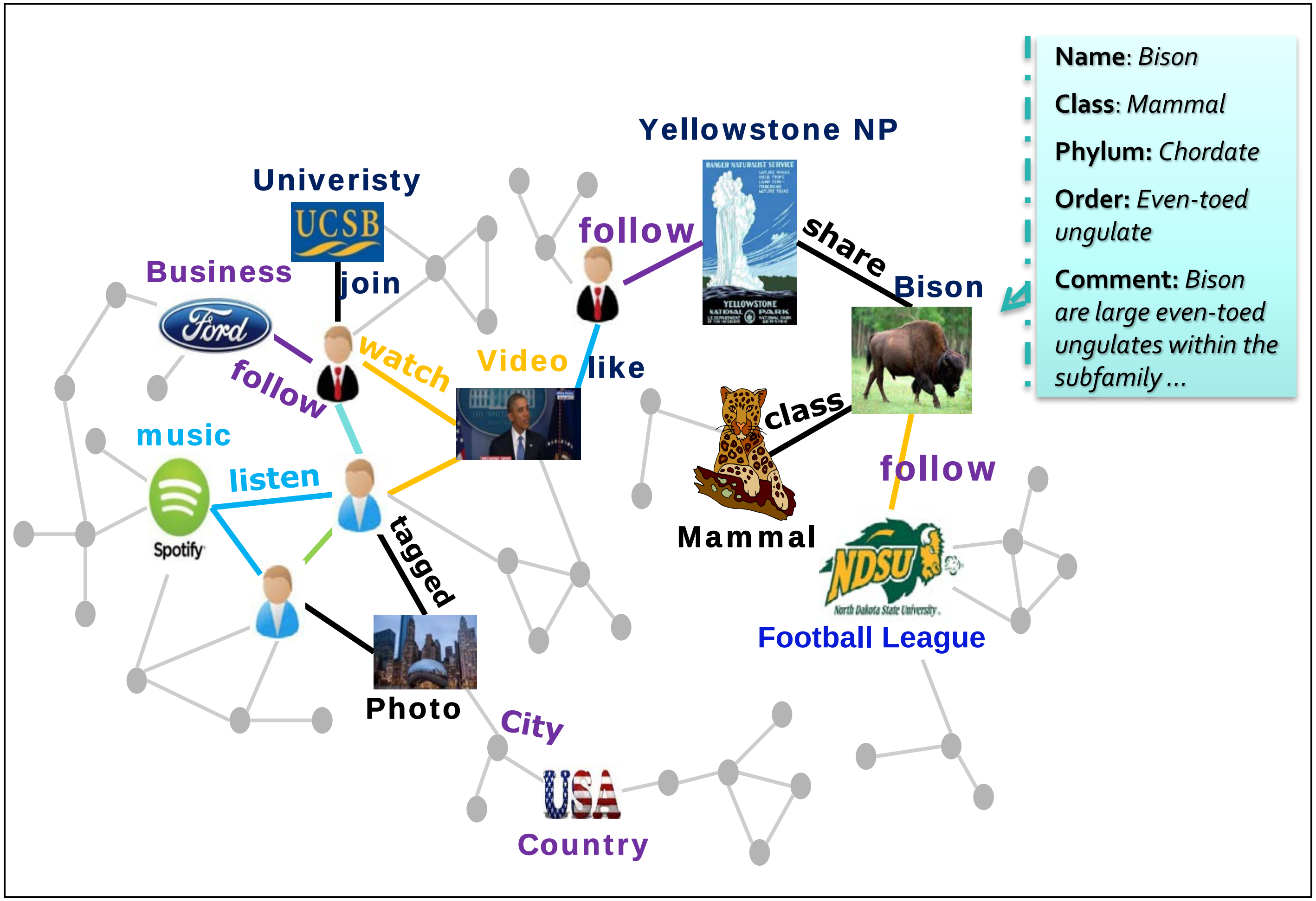


Question Answering History



- Big & linked data
- QA over linked data
  - Addresses practical problems of data accessibility in a data heterogeneity scenario.

Blossom of Large-scale Knowledge Bases



DBpedia

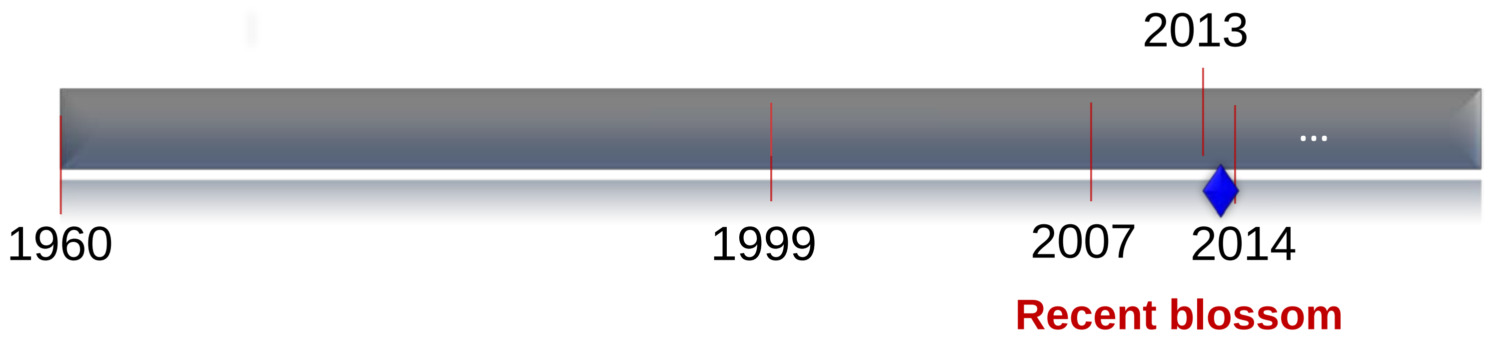
yago select knowledge

Freebase

SATORI

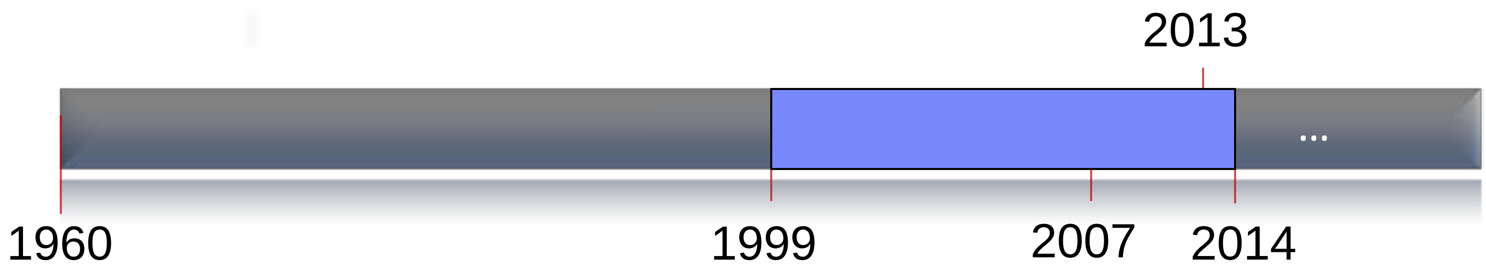
Courtesy of Shengqi Yang, UCSB

Question Answering History



- Data source: Knowledge bases & the Web
- Different Methodologies
  - Semantic parsing
  - Graph querying
  - Open IE and paraphrase
  - Embedding methods
  - Feature-based

CS290D covers



- Open domain QA (TREC, 1999~2007) (April 15th)
- QA over linked data (~2007-) (April 15th)
- Recent Developments (~2013-) (April 29th)

Preliminaries

Terminology: Question Phrase

The part of a question that says what is being asked:

- Wh-words:
  - who, what, which, when, where, why, and how
- Wh-words + nouns, adjectives or adverbs:
  - “which party …”, “which actress …”, “how long …”, “how tall …”.

Terminology: Question Type

Useful for distinguishing different processing strategies

- **FACTOID**:
  - **PREDICATIVE QUESTIONS:**
    - “Who was the first man in space?”
    - “What is the highest mountain in Korea?”
    - “How far is Earth from Mars?”
    - “When did the Jurassic Period end?”
    - “Where is Taj Mahal?”
  - **LIST**:
    - “Give me all cities in Germany.”
  - **SUPERLATIVE**:
    - “What is the highest mountain?”
  - **YES-NO**:
    - “Was Margaret Thatcher a chemist?”

Terminology: Question Type

Useful for distinguishing different processing strategies

- **OPINION**:
  - "What do most Americans think of gun control?"
- **CAUSE & EFFECT**:
  - "What is the most frequent cause for lung cancer?"
- **PROCESS:**
  - "How do I make a cheese cake?"
- **EXPLANATION & JUSTIFICATION:**
  - "Why did the revenue of IBM drop?"
- **ASSOCIATION QUESTION:**
  - "What is the connection between Barack Obama and Indonesia?"
- **EVALUATIVE OR COMPARATIVE QUESTIONS:**
  - "What is the difference between impressionism and expressionism?"

Terminology: Answer Type

- The class of object sought by the question:
  - **Abbreviation**
  - **Entity:** event, color, animal, plant,. . .
  - **Description, Explanation & Justification :** definition, manner, reason,. . . (“How, why …”)
  - **Human:** group, individual,. . . (“Who …”)
  - **Location:** city, country, mountain,. . . (“Where …”)
  - **Numeric:** count, distance, size,. . . (“How many how far, how long …”)
  - **Temporal:** date, time, …(from “When …”)

Terminology: Question Focus & Topic

- Question focus is the **property** or **entity** that is being sought by the question
  - “In which **city** was Barack Obama born?”
  - “What is the **population** of Galway?”

- Question topic: What the question is generally about :
  - “What is the height of Mount Everest?”
    - (geography, mountains)
  - “Which organ is affected by the Meniere’s disease?”
    - (medicine)

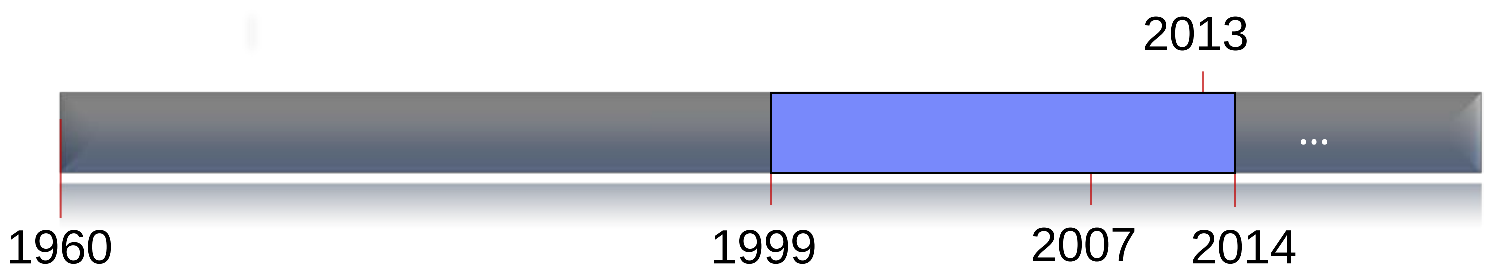
Common Evaluation Metrics

1. Accuracy (does answer match gold-labeled answer?)

2. Mean Reciprocal Rank (MRR)
   - For each query return a ranked list of M candidate answers.
   - Query score is 1/Rank of the first correct answer
     - *If first answer is correct: 1*
     - *else if second answer is correct: ½*
     - *else if third answer is correct: ⅓, etc.*
     - *Score is 0 if none of the M answers are correct*
   - Take the mean over all N queries

$$MRR = \frac{\sum_{i=1}^{N} \frac{1}{rank_i}}{N}$$

CS290D covers

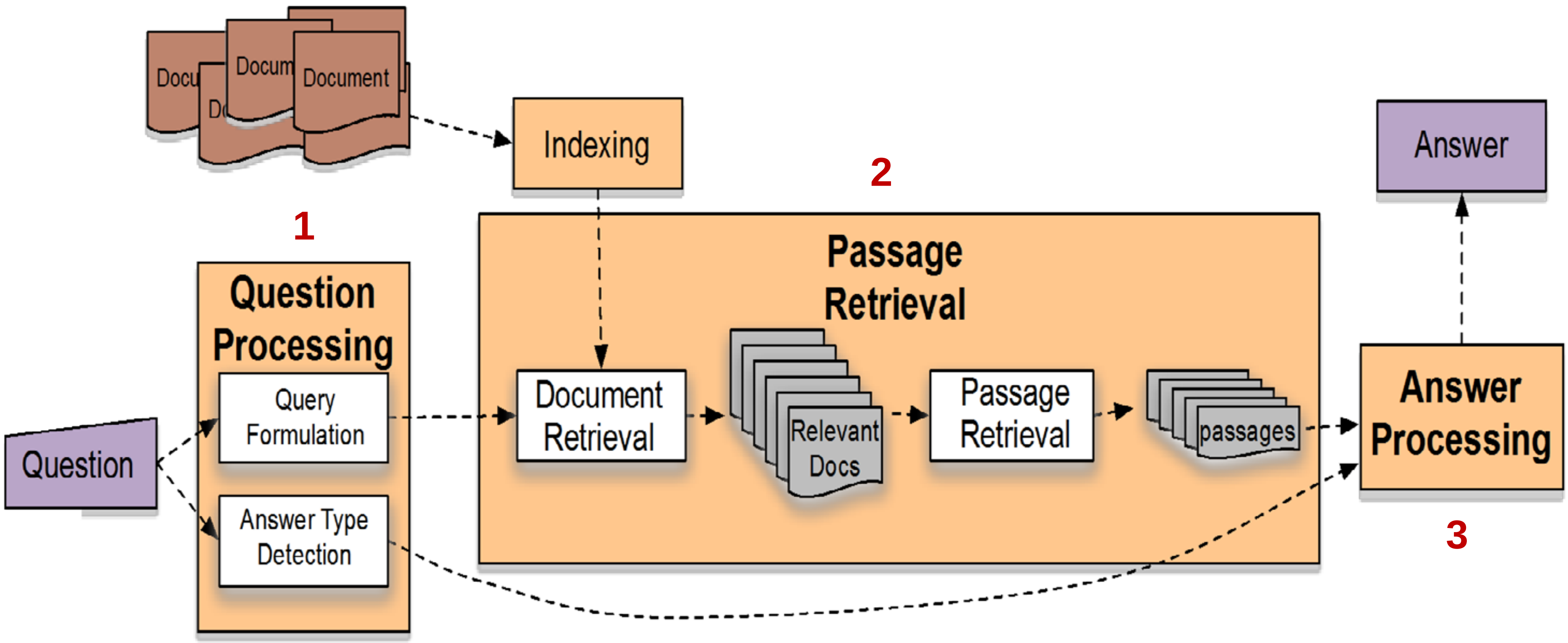


- **Open domain QA (TREC, 1999~2007)**
- QA over linked data (~2007-)
- Recent Developments (~2013-)

TREC, 1999~2007

Main Methodology

Information Retrieval (IR) based QA



IR-based QA

1. QUESTION PROCESSING

- Detect question type, answer type, focus, relations
- Formulate queries to send to a search engine

2. PASSAGE RETRIEVAL

- Retrieve ranked documents
- Break into suitable passages and re-rank

3. ANSWER PROCESSING

- Extract candidate answers
- Rank candidates
  - using evidence from the text and external sources

Question Processing

- Answer Type Detection
  - Decide the **named entity type** (person, place) of the answer
- Query Formulation
  - Choose **query keywords** for the IR system
- Question Type Classification
  - Is this a definition question, a math question, a list question?
- Focus Detection
  - Find the question words that are replaced by the answer
- Relation Extraction
  - Find relations between entities in the question

Question Processing

What are the two states you could be re-entering if you’re crossing Florida’s northern border?

- Answer Type: US state
- Query Formulation: two states, border, Florida, north
- Question Focus: the two states
- Relations: borders(Florida, ?x, north)

Answer Type Detection

- *Who founded Virgin Airlines?*
  - PERSON
- *What Canadian city has the largest population?*
  - *CITY.*

Answer Type Taxonomy

Xin Li, Dan Roth. 2002. Learning Question Classifiers. COLING'02

- 6 coarse classes
  - ABBEVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION, NUMERIC
- 50 finer classes
  - LOCATION: city, country, mountain…
  - HUMAN: group, individual, title, description
  - ENTITY: animal, body, color, currency…

How to Detect Answer Types?

- Hand-written rules
- Machine Learning
- Hybrids

How to Detect Answer Types?

- Hand-written rules
  - Regular expression-based rules can get some cases:
    - Who {is|was|are|were} PERSON
    - Where {is|was|are|were} LOCATION
  - Other rules use the **question headword**:

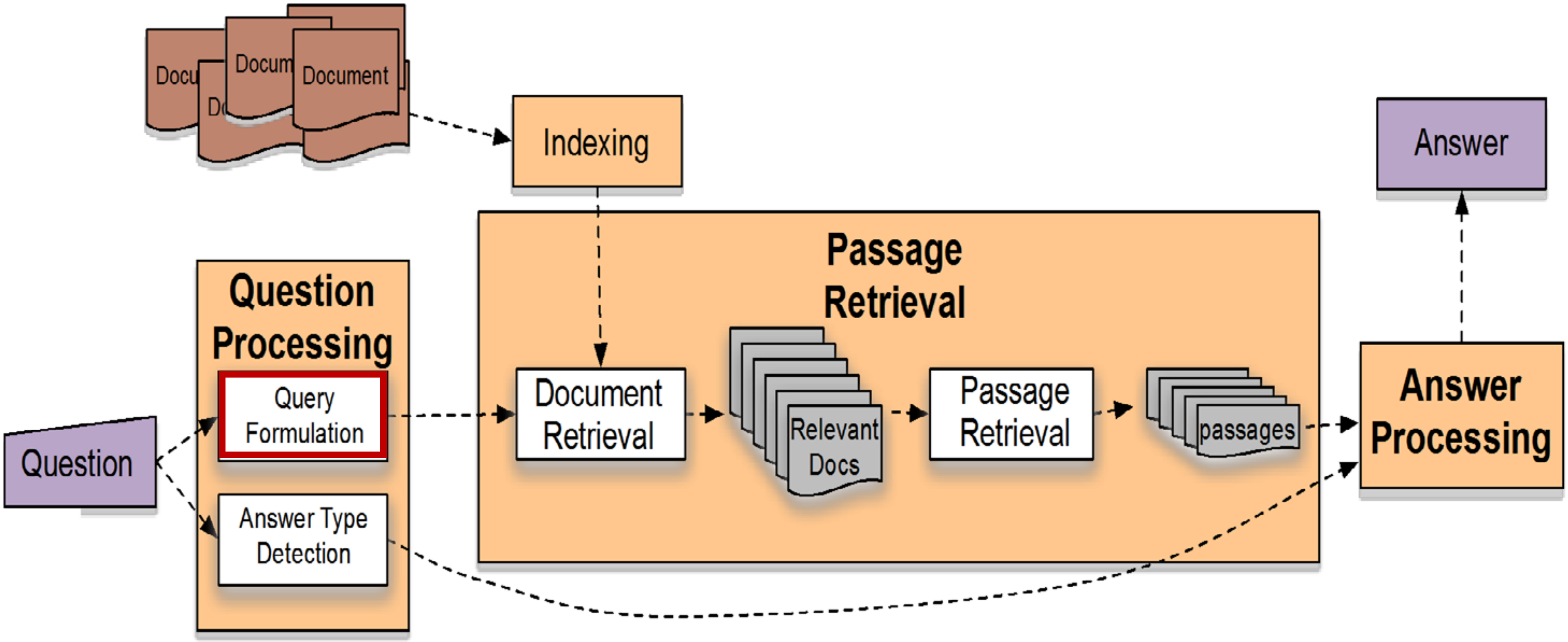
    (the headword of the first noun phrase after the wh-word)
    - Which **city** in China has the largest number of foreign financial companies?
    - What is the state **flower** of California?

How to Detect Answer Types?

- Treat the problem as machine learning classification
  - **Define** a taxonomy of question types
  - **Annotate** training data for each question type
  - **Train** classifiers for each question class using a rich set of features.
    - features include those based on hand-written rules!

e.g.,
- Question words and phrases
- Part-of-speech tags
- Parse features (headwords)
- Named entities
- Semantically related words

IR-based QA

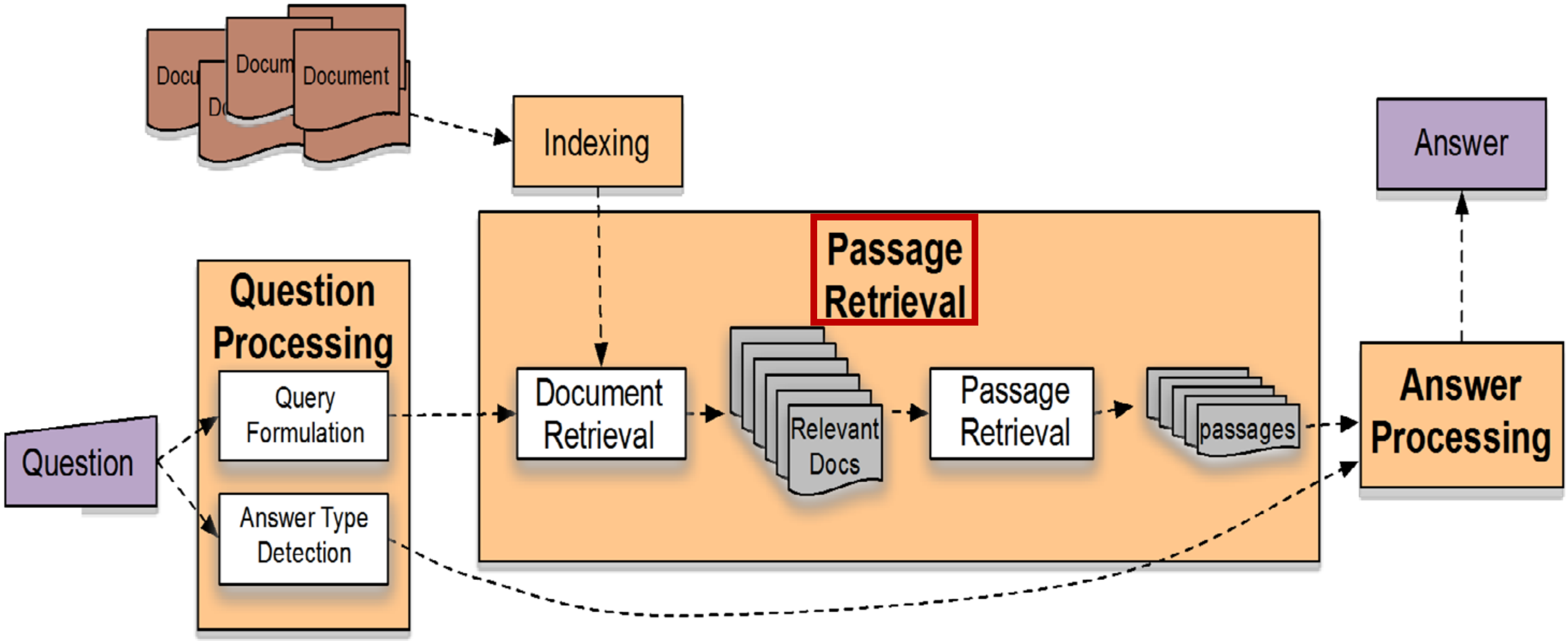

Document
Docu
Docum
Document
Indexing
Answer
Passage Retrieval
Question Processing
Query Formulation
Answer Type Detection
Question
Document Retrieval
Relevant Docs
Passage Retrieval
passages
Answer Processing

Keyword Selection Algorithm

1. Select all non-stop words in quotations
2. Select all NNP words in recognized named entities
3. Select all complex nominals with their adjectival modifiers
4. Select all other complex nominals
5. Select all nouns with their adjectival modifiers
6. Select all other nouns
7. Select all verbs
8. Select all adverbs
9. Select the QFW word (skipped in all previous steps)
10. Select all other words

Dan Moldovan et al., Proceedings of TREC-8., 1999.

IR-based QA


Document
Indexing
Answer
Passage Retrieval
Question Processing
Query Formulation
Answer Type Detection
Question
Document Retrieval
Relevant Docs
Passage Retrieval
passages
Answer Processing

Passage Retrieval

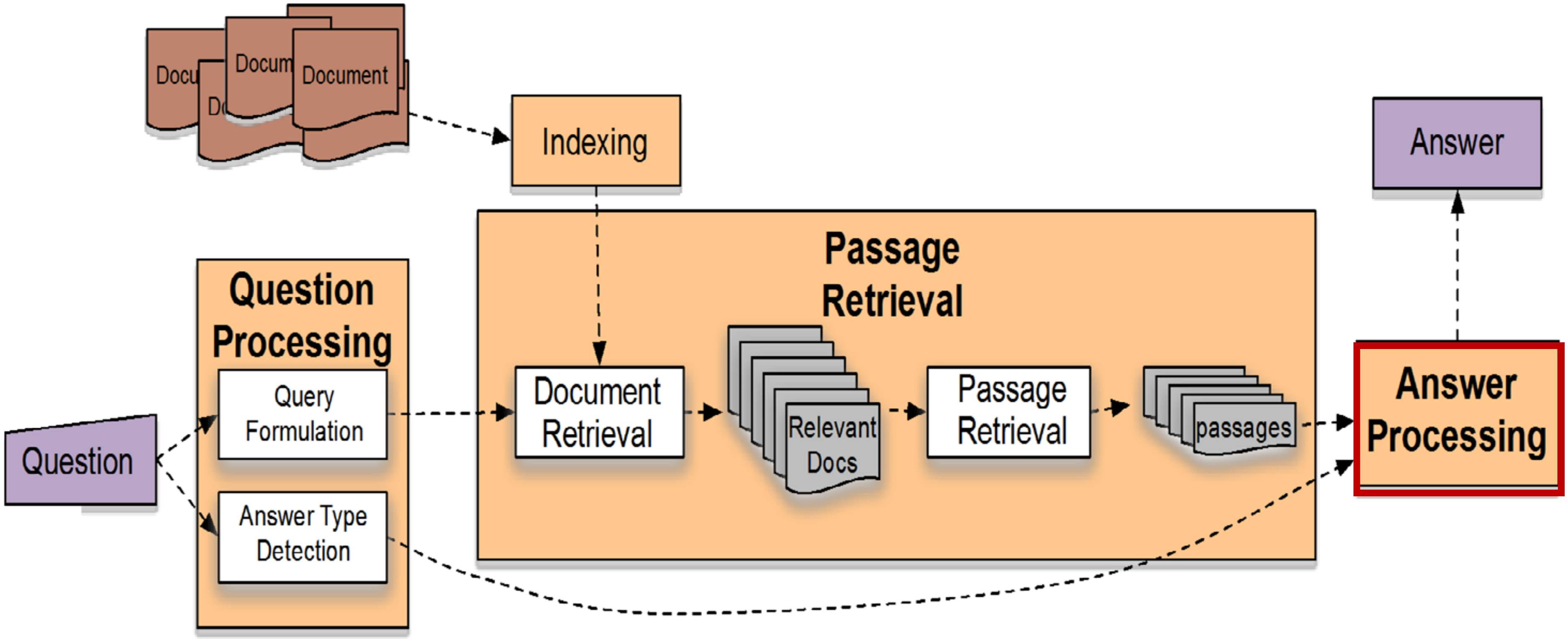
- Step 1: IR engine retrieves documents using query terms
- Step 2: Segment the documents into shorter units
  - e.g., paragraphs
- Step 3: Passage ranking

Features for Passage Ranking

Either in rule-based classifiers or with supervised machine learning

- Number of Named Entities of the right type in passage
- Number of query words in passage
- Number of question N-grams also in passage
- Proximity of query keywords to each other in passage
- Longest sequence of question words
- Rank of the document containing passage

IR-based QA


Document
Docum
Docu
Do
Indexing
Answer
Passage Retrieval
Question Processing
Query Formulation
Answer Type Detection
Question
Document Retrieval
Relevant Docs
Passage Retrieval
passages
Answer Processing

Answer Extraction

- Run an answer-type named-entity tagger on the passages
  - Each answer type requires a named-entity tagger that detects it
  - If answer type is CITY, tagger has to tag CITY
    - Can be full NER, simple regular expressions, or hybrid
- Return the string with the right type:
  - Who is the prime minister of India (PERSON)

    Manmohan Singh, Prime Minister of India, had told left leaders that the deal would not be renegotiated.
  - How tall is Mt. Everest? (LENGTH)

    The official height of Mount Everest is **29035 feet**

Ranking Candidate Answers

- But what if there are multiple candidate answers!

  Q: Who was Queen Victoria's second son?

  Answer Type: **Person**

- Passage:

  The Marie biscuit is named after Marie Alexandrovna, the daughter of Czar Alexander II of Russia and wife of Alfred, the second son of Queen Victoria and Prince Albert

Ranking Candidate Answers

- But what if there are multiple candidate answers!

  Q: Who was Queen Victoria's second son?

  Answer Type: **Person**

- Passage:

The Marie biscuit is named after **Marie Alexandrovna**, the daughter of **Czar Alexander II of Russia** and wife of **Alfred**, the second son of **Queen Victoria** and **Prince Albert**

Use machine learning: Features for ranking candidate answers

Answer type match: Candidate contains a phrase with the correct answer type.

Pattern match: Regular expression pattern matches the candidate.

Question keywords: # of question keywords in the candidate.

Keyword distance: Distance in words between the candidate and query keywords

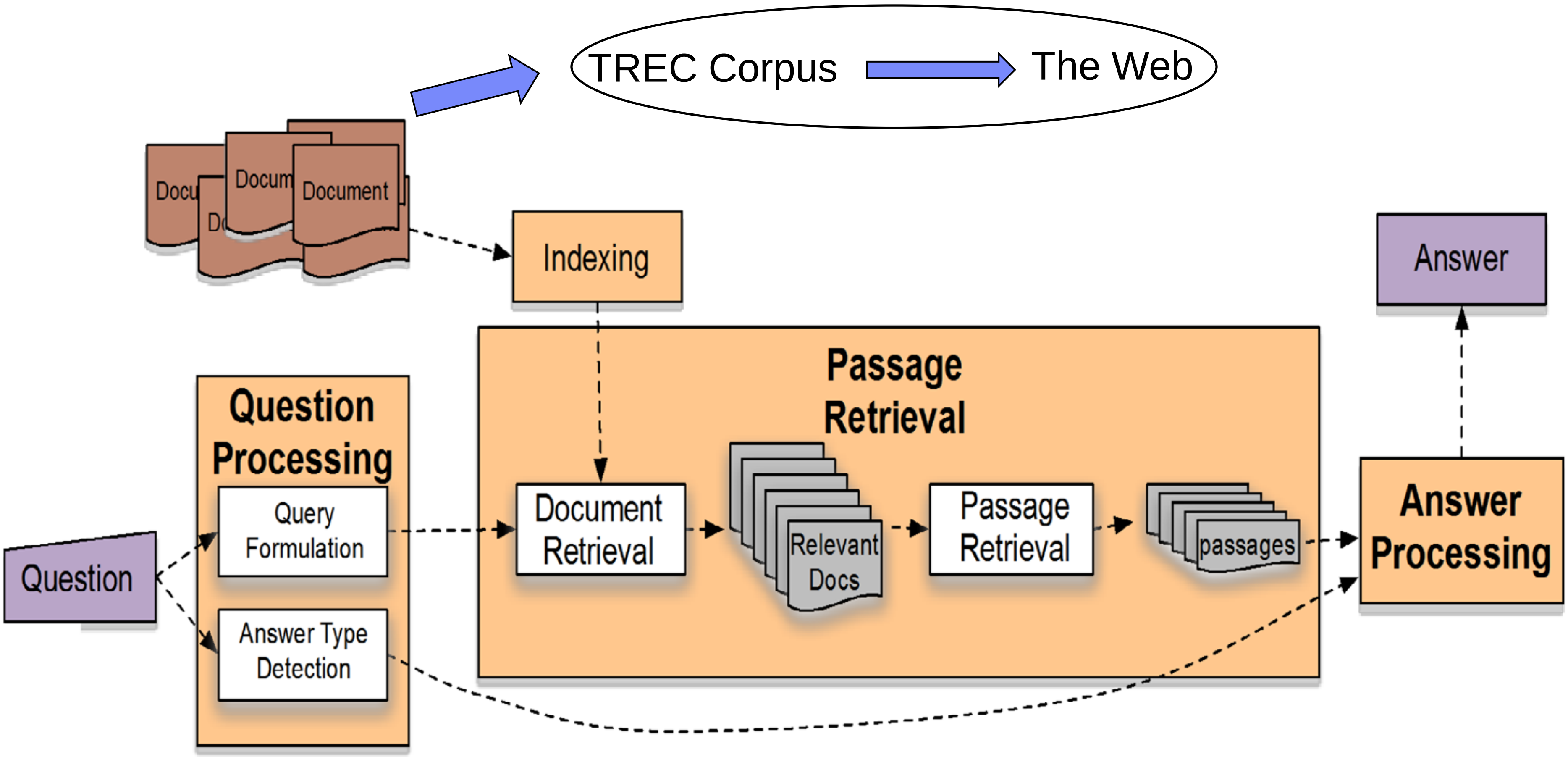
Novelty factor: A word in the candidate is not in the query.

Apposition features: The candidate is an appositive to question terms

Punctuation location: The candidate is immediately followed by a comma, period, quotation marks, semicolon, or exclamation mark.

Sequences of question terms: The length of the longest sequence of question terms that occurs in the candidate answer.

IR-based QA: Review

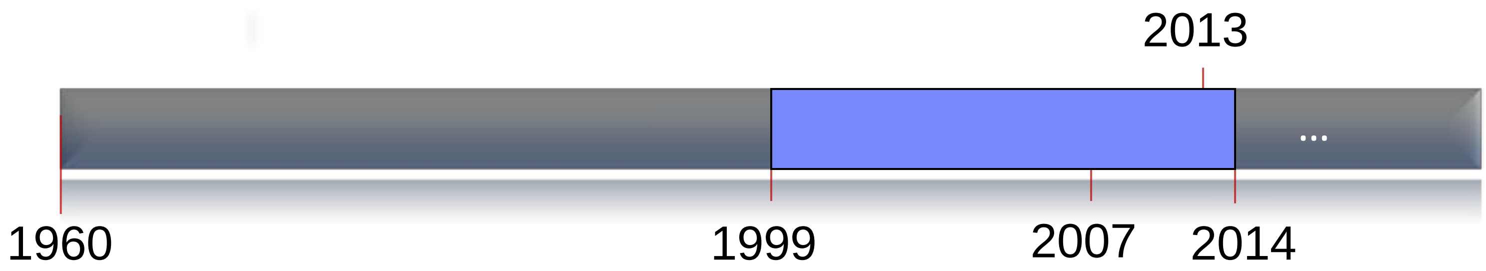


Top Performing Systems

- Currently the best performing systems at TREC can answer approximately 70% of the questions
- Approaches and successes have varied a fair deal
  - Knowledge-rich approaches, using a vast array of NLP techniques stole the show in 2000, 2001, still do well
    - Notably Harabagiu, Moldovan et al. – SMU/UTD/LCC
  - AskMSR system stressed how much could be achieved by very simple methods with enough text (and now various copycats) **(TREC Corpus => the Web)**
  - Middle ground is to use large collection of surface matching patterns (ISI)

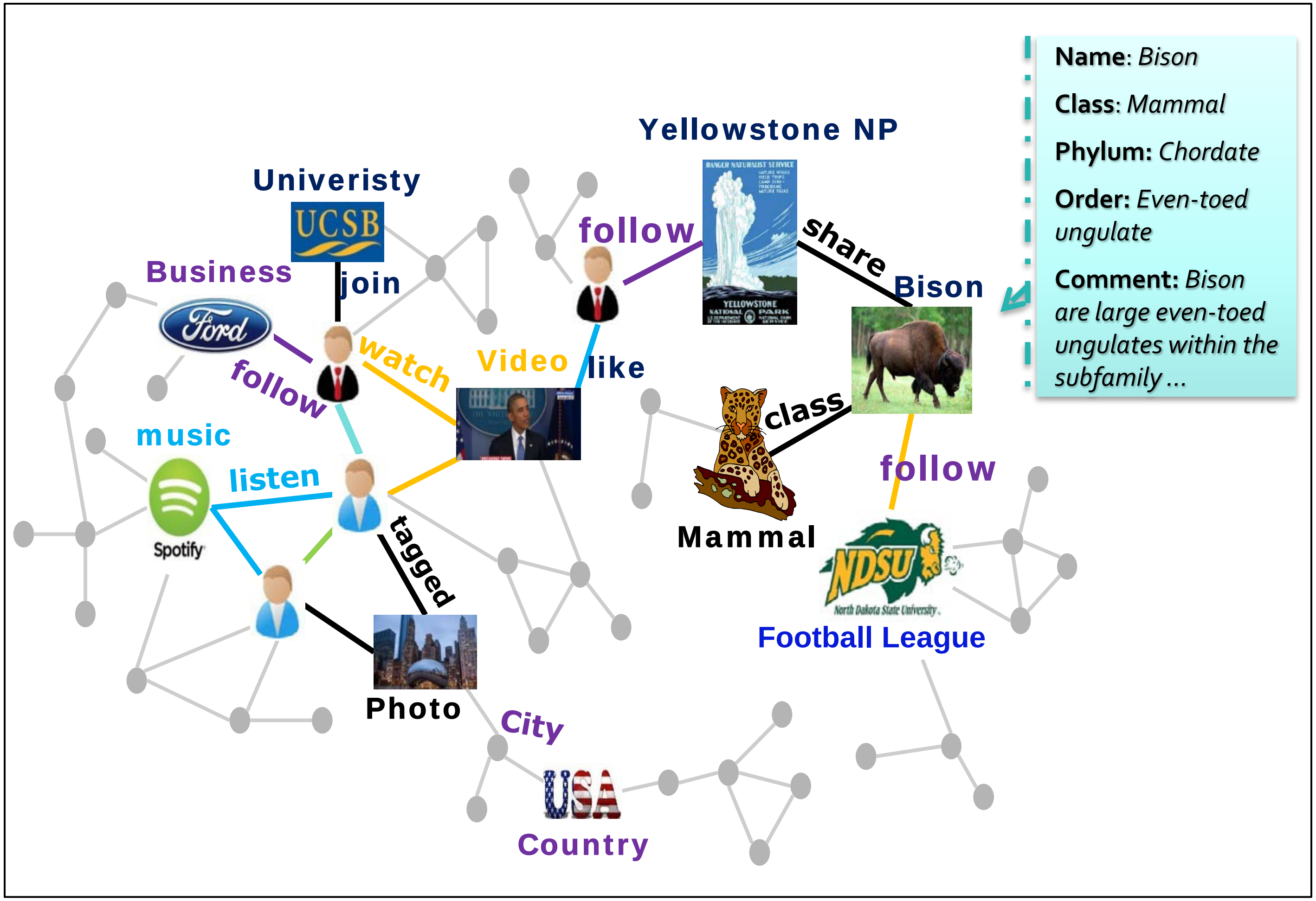
Dan Klein, Spring 2010

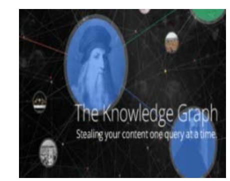
CS290D covers



- Open domain QA (TREC, 1999~2007)
- **QA over linked data (~2007-)**
- Recent developments (~2013-)

Blossom of Large-scale Knowledge Bases





Courtesy of Shengqi Yang, UCSB

From Rigid Schemas to Rich Schemas

- Heterogeneous, complex and large-scale databases.
- Very-large and dynamic "schemas".

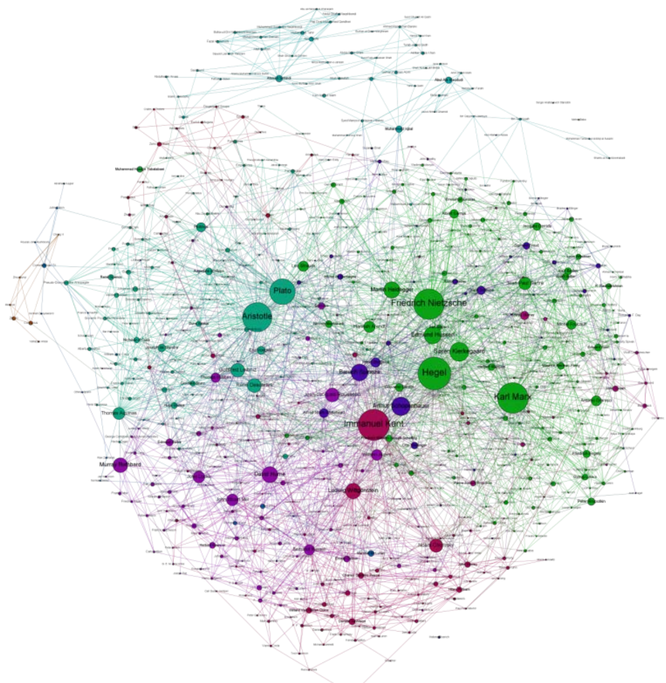
circa 2013

1,000s-1,000,000s attributes

circa 2000

10s-100s attributes

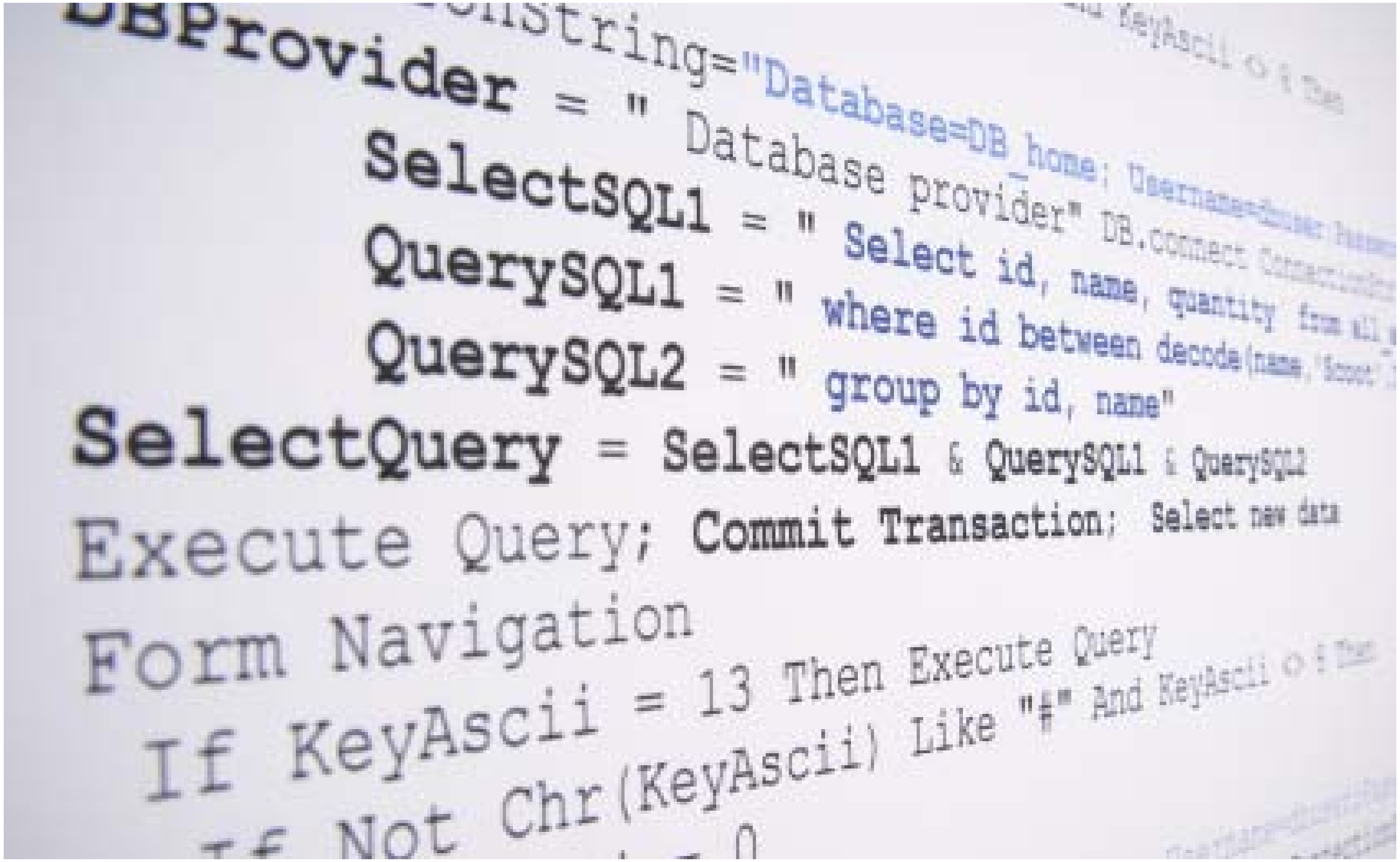
EMP_NO	FIRST_NAME	LAST_NAME	PHONE_EXT	HIRE_DATE	DEPT...	JOB_C...	JOB_GR...	JOB_COUNT...	SALARY	FULL_NAME
2	Robert	Nelson	250	12.28.1988 12:00 am	600	VP	2	USA	105.900,00	Nelson, Robert
4	Bruce	Young	233	12.28.1988 12:00 am	621	Eng	2	USA	97.500,00	Young, Bruce
5	Kim	Lambert	22	02.06.1989 12:00 am	130	Eng	2	USA	102.750,00	Lambert, Kim
8	Leslie	Johnson	410	04.05.1989 12:00 am	180	Mktg	3	USA	64.635,00	Johnson, Leslie
9	Phil	Forest	229	04.17.1989 12:00 am	622	Mngr	3	USA	75.060,00	Forest, Phil
11	K. J.	Weston	34	01.17.1990 12:00 am	130	SRep	4	USA	86.292,94	Weston, K. J.
12	Terri	Lee	256	05.01.1990 12:00 am	000	Admin	4	USA	53.793,00	Lee, Terri
14	Stewart	Hall	227	06.04.1990 12:00 am	900	Finan	3	USA	69.482,63	Hall, Stewart
15	Katherine	Young	231	06.14.1990 12:00 am	623	Mngr	3	USA	67.241,25	Young, Katherine
20	Chris	Papadopoulos	887	01.01.1990 12:00 am	671	Mngr	3	USA	89.655,00	Papadopoulos, Chi
24	Pete	Fisher	888	09.12.1990 12:00 am	671	Eng	3	USA	81.810,19	Fisher, Pete
28	Ann	Bennet	5	02.01.1991 12:00 am	120	Admin	5	England	22.935,00	Bennet, Ann
29	Roger	De Souza	288	02.18.1991 12:00 am	623	Eng	3	USA	69.482,63	De Souza, Roger
34	Janet	Baldwin	2	03.21.1991 12:00 am	110	Sales	3	USA	61.637,81	Baldwin, Janet
36	Roger	Reeves	6	04.25.1991 12:00 am	120	Sales	3	England	33.620,63	Reeves, Roger
37	Willie	Stansbury	7	04.25.1991 12:00 am	120	Eng	4	England	39.224,06	Stansbury, Willie
44	Leslie	Phong	216	06.03.1991 12:00 am	623	Eng	4	USA	56.034,38	Phong, Leslie
45	Ashok	Ramanathan	209	08.01.1991 12:00 am	621	Eng	3	USA	80.689,50	Ramanathan, Ashc
46	Walter	Steadman	210	08.09.1991 12:00 am	900	CFO	1	USA	116.100,00	Steadman, Walter
52	Carol	Nordstrom	420	10.02.1991 12:00 am	180	PRel	4	USA	42.742,50	Nordstrom, Carol
61	Luke	Leung	3	02.18.1992 12:00 am	110	SRep	4	USA	68.805,00	Leung, Luke
65	Sue Anne	O'Brien	877	03.23.1992 12:00 am	670	Admin	5	USA	31.275,00	O'Brien, Sue Anne



[Andrei Freitas, 2013]

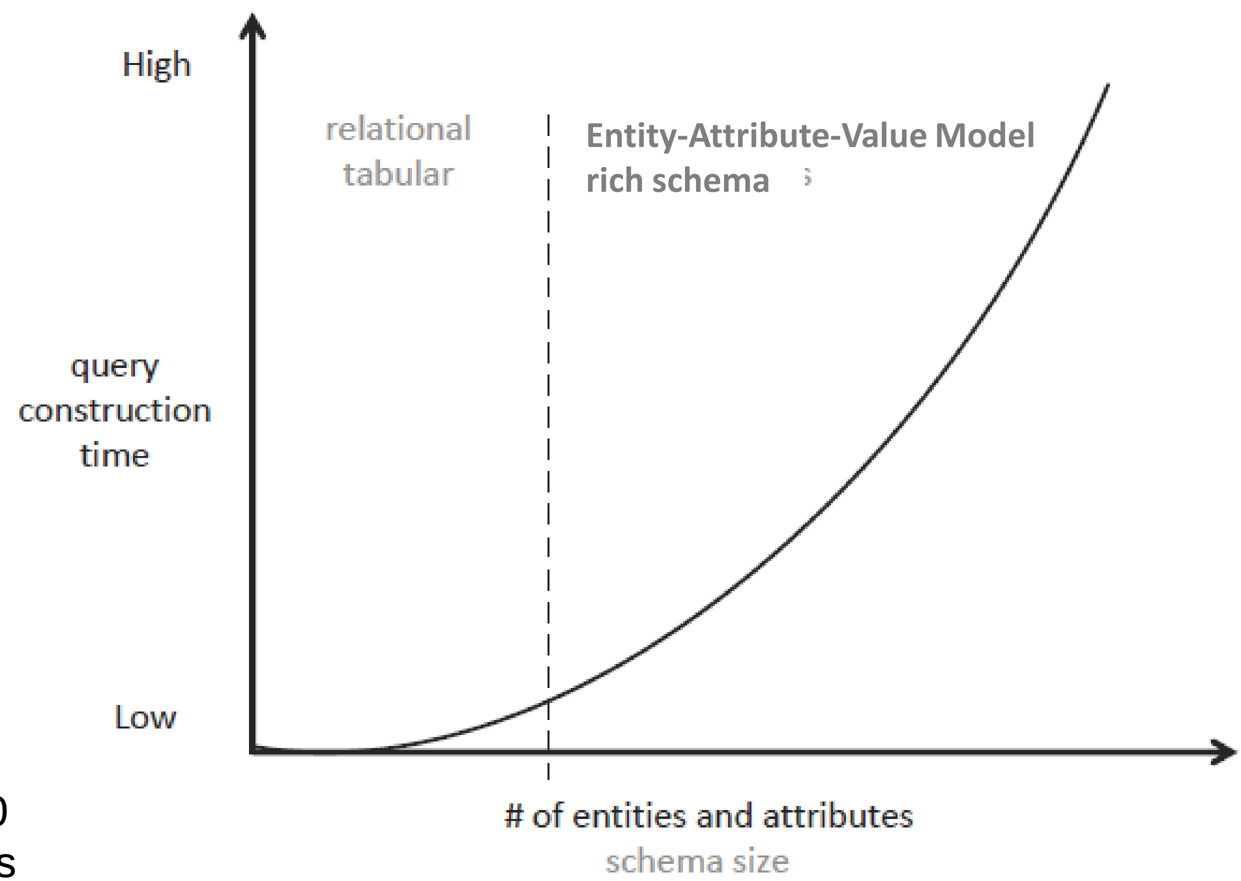
Big & Linked Data: Structured queries

- Big Problem: Structured queries are still the primary way to query databases.



Big & Linked Data: Structured queries

- Query construction time vs schema size



~10-100 attributes

~10^3-10^6 attributes

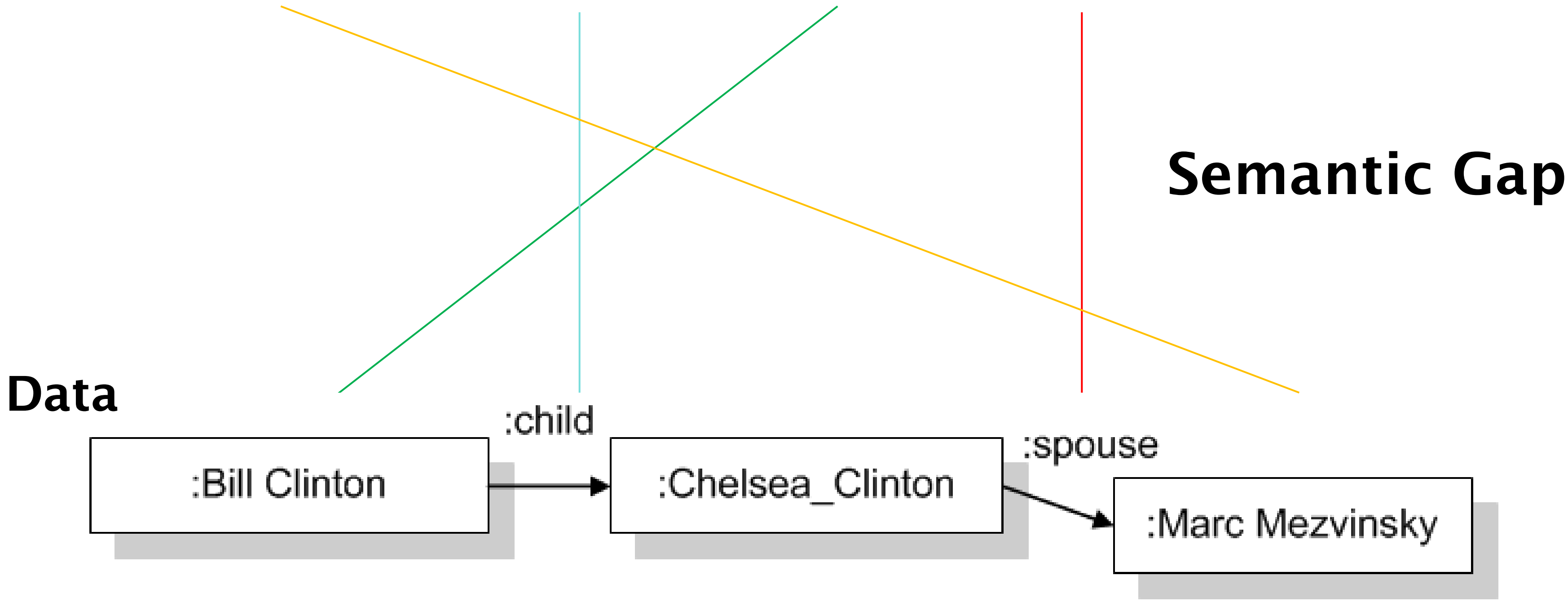
[Andrei Freitas, 2013]

QA over Linked Data

- Schema size and heterogeneity represent a fundamental shift for databases.
- Addressing the associated data management challenges (specially querying) depends on the development of principled semantic models for databases.
- QA/Natural Language Interfaces (NLIs) as schema-agnostic query mechanisms.

Vocabulary Problem for Databases

Query: Who is the daughter of Bill Clinton married to ?



QA Systems over Linked Data (before 2013)
Bridge the semantic gap

- Aqualog & PowerAqua (Lopez et al. 2006)
  - Querying on a Semantic Web scale
- ORAKEL & Pythia (Cimiano et al, 2007; Unger & Cimiano, 2011)
  - Ontology-specific question answering
- IBM Watson (Ferrucci et al., 2010)
  - Large-scale evidence-based model for QA
- Freya (Damljanovic et al., 2010)
- Treo (Freitas et al., QALD-1@ESWC 2011)
  - Schema-agnostic querying using distributional semantics
- Deanna (Yahya et al., 2012)
- **TBSL (Unger et al., WWW 2012)**
  - **Template-based question answering**

TBSL (Unger et al., WWW 2012)

Motivation

- In order to understand a user question, we need to understand:

The words (dataset-specific)

Abraham Lincoln → res:Abraham Lincoln

died in → dbo:deathPlace

The semantic structure (dataset-independent)

who → SELECT ?x WHERE { … }

the most *N* → ORDER BY DESC(COUNT(?N)) LIMIT 1

more than *i N* → HAVING COUNT(?N) > i

TBSL (Unger et al., WWW 2012)

Key contributions:

1. Constructs a query template that directly mirrors the linguistic structure of the question

Who produced the most films? →

SPARQL template:

```
SELECT DISTINCT ?x WHERE {
    ?x ?p ?y .
    ?y rdf:type ?c .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
  ?c CLASS [films]
  ?p PROPERTY [produced]
```

2. Instantiates the template by matching natural language expressions with ontology concepts

SPARQL template:

```
SELECT DISTINCT ?x WHERE {
    ?x ?p ?y .
    ?y rdf:type ?c .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
  ?c CLASS [films]
  ?p PROPERTY [produced]
```

→

SPARQL query:

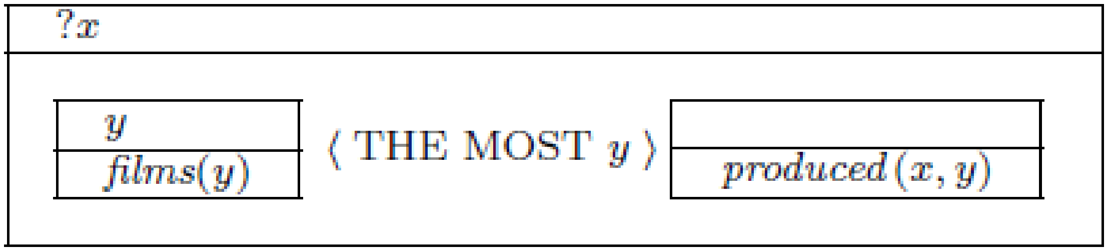
```
SELECT DISTINCT ?x WHERE {
  ?x <http://dbpedia.org/ontology/producer> ?y .
  ?y rdf:type <http://dbpedia.org/ontology/Film> .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
```

Step 1: Template generation

Linguistic processing

Example:
Who produced the most films?

Pythia parser [Unger et al., NLDB'2011]



Rules/heuristics, e.g.,
1. Nouns often referring to classes and properties
2. Verbs often referring to properties

SPARQL template 1:

```
SELECT DISTINCT ?x WHERE {
    ?x ?p ?y .
    ?y rdf:type ?c .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
  ?c CLASS [films]
  ?p PROPERTY [produced]
```

SPARQL template 2:

```
SELECT DISTINCT ?x WHERE {
    ?x ?p ?y .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
    ?p PROPERTY [films]
```

Step 2: Template instantiation

Entity identification and predicate detection

Example:
Who produced the most films?

SPARQL template 1:

```
SELECT DISTINCT ?x WHERE {
      ?x ?p ?y .
      ?y rdf:type ?c .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
   ?c CLASS [films]
   ?p PROPERTY [produced]
```

?c CLASS [films]

<http://dbpedia.org/ontology/Film>
<http://dbpedia.org/ontology/FilmFestival>
...

?p PROPERTY [produced]

<http://dbpedia.org/ontology/producer>
<http://dbpedia.org/property/producer>
<http:// dbpedia.org/ontology/wineProduced>

Instantiation

```
SELECT DISTINCT ?x WHERE {
  ?x <http://dbpedia.org/ontology/producer> ?y .
  ?y rdf:type <http://dbpedia.org/ontology/Film>
.
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
```

Step 3: Query ranking and selection

Example:
Who produced the most films?

```
SELECT DISTINCT ?x WHERE {
  ?x <http://dbpedia.org/ontology/producer> ?y .
  ?y rdf:type <http://dbpedia.org/ontology/Film> .
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
```

Score: 0.76

```
SELECT DISTINCT ?x WHERE {
  ?x <http://dbpedia.org/ontology/producer> ?y .
  ?y rdf:type <http://dbpedia.org/ontology/FilmFestival>.
}
ORDER BY DESC(COUNT(?y)) LIMIT 1
```

Score: 0.60

Main drawback

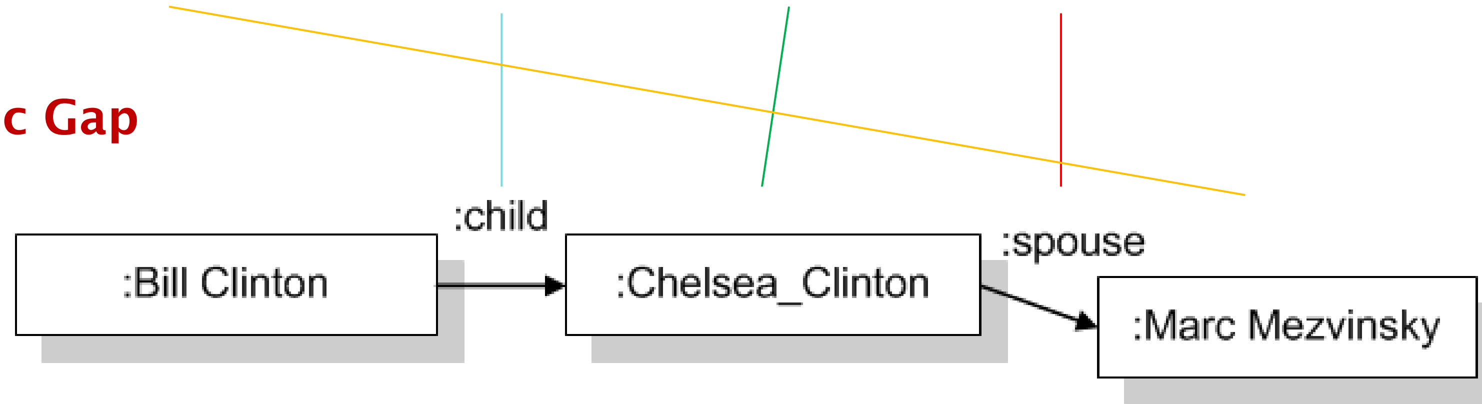
- Manual rules to generate query template.

- Considering all possibilities of how the data could be modelled leads to a big amount of templates (and even more queries) for one question.

Question: Who is the daughter of Bill Clinton married to ?

Semantic Gap

Data:



Other Methodologies to Bridge the Gap

1. Semantic Parsing
2. Graph Querying
3. Embedding-based
4. …

To be discussed on April 29th

Possible QA-related Projects

- Deep learning framework for QA
  - Convolutional Neural Networks
  - Recurrent Neural Networks (directly generate answer!)
- Incorporating semantics in resolving your task
  - Not only question answering! But general text analysis
  - Word embedding, WordNet
  - Knowledge provided in DBpedia, Freebase, YAGO etc.
- Contextual QA and dialogue systems
  - Instant feedback
  - More evidence
- Domain specific question answering
  - Discussions in all kinds of forums
  - Domain knowledge

Do-it-yourself (DIY): Core Resources

Datasets

- DBpedia
  - http://dbpedia.org/
- YAGO
  - http://www.mpi-inf.mpg.de/yago-naga/yago/
- Freebase
  - http://www.freebase.com/
- Wikipedia dumps
  - http://dumps.wikimedia.org/
- ConceptNet
  - http:// conceptnet5.media.mit.edu/
- Common Crawl
  - http://commoncrawl.org/

- Where to use:
  - As a commonsense KB or as a data source

Wikipedia

- High domain coverage:
  - ~95% of Jeopardy! Answers.
  - ~98% of TREC answers.
- Wikipedia is entity-centric.
- Curated link structure.
- Complementary tools:
  - Wikipedia Miner
- Where to use:
  - Construction of distributional semantic models.
  - As a commonsense KB

Lexical Resources

- WordNet
  - http://wordnet.princeton.edu/
- Wiktionary
  - http://www.wiktionary.org/
  - API: https://www.mediawiki.org/wiki/API:Main_page
- FrameNet
  - https://framenet.icsi.berkeley.edu/fndrupal/
- VerbNet
  - http://verbs.colorado.edu/~mpalmer/projects/verbnet.html
- English lexicon for DBpedia 3.8 (in the lemon format)
  - http://lemon-model.net/lexica/dbpedia_en/
- PATTY (collection of semantically-typed relational patterns)
  - http://www.mpi-inf.mpg.de/yago-naga/patty/
- BabelNet
  - http://babelnet.org/

- Where to use:
  - Query expansion
  - Semantic similarity
  - Semantic relatedness
  - Word sense disambiguation

Indexing & Search Engines

- Lucene & Solr
  - http://lucene.apache.org/
- Terrier
  - http://terrier.org/
- Where to use:
  - Answer Retrieval
  - Scoring
  - Query-Data matching

Text Processing Tools

- GATE (General Architecture for Text Engineering)
  - http://gate.ac.uk/
- NLTK (Natural Language Toolkit)
  - http://nltk.org/
- Stanford NLP
  - http://www-nlp.stanford.edu/software/index.shtml
- LingPipe
  - http://alias-i.com/lingpipe/index.html
- Where to use:
  - Question Analysis

Parsers

- MALT
  - http://www.maltparser.org/
  - Languages (pre-trained): English, French, Swedish
- Stanford parser
  - http://nlp.stanford.edu/software/lex-parser.shtml
  - Languages: English, German, Chinese, and others
- CHAOS
  - http://art.uniroma2.it/external/chaosproject/
  - Languages: English, Italian
- C&C Parser
  - http://svn.ask.it.usyd.edu.au/trac/candc
- Where to Use:
  - Question Analysis

Named Entity Recognition/Linking

- NERD (Named Entity Recognition and Disambiguation)
  - http://nerd.eurecom.fr/
- Stanford Named Entity Recognizer
  - http://nlp.stanford.edu/software/CRF-NER.shtml
- FOX (Federated Knowledge Extraction Framework)
  - http://fox.aksw.org
- DBpedia Spotlight
  - http://spotlight.dbpedia.org
- Where to use:
  - Question Analysis
  - Query-Data Matching

String Similarity and Semantic Relatedness

- Wikipedia Miner
  - http://wikipedia-miner.cms.waikato.ac.nz/
- WS4J (Java API for several semantic relatedness algorithms)
  - https://code.google.com/p/ws4j/
- SecondString (string matching)
  - http://secondstring.sourceforge.net
- EasyESA (distributional semantic relatedness)
  - http://treo.deri.ie/easyESA
- Sspace (distributional semantics framework)
  - https://github.com/fozziethebeat/S-Space
- Where to use:
  - Query-Data matching
  - Semantic relatedness & similiarity
  - Word Sense Disambiguation

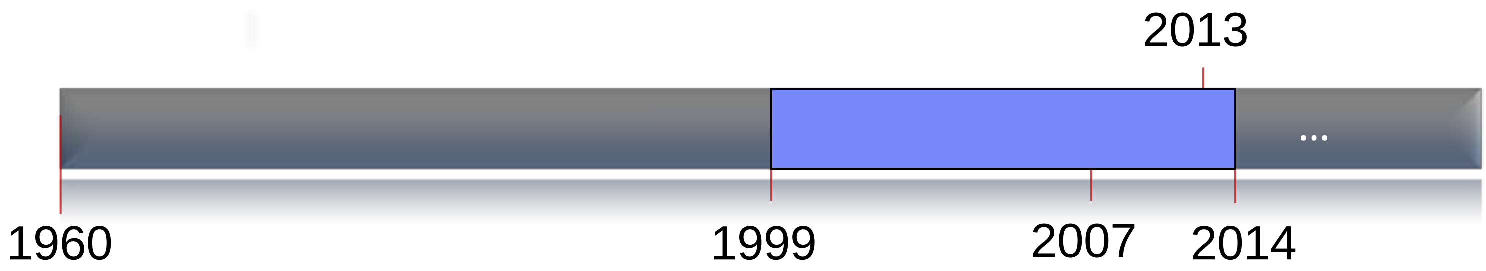
Text Entailment

- DIRT
  - Paraphrase Collection:
    - http://aclweb.org/aclwiki/index.php?title
  - DIRT_Paraphrase_Collection
    - Demo:
      - http://demo.patrickpantel.com/demos/lexsem/paraphrase.htm
- PPDB (The Paraphrase Database)
  - http://www.cis.upenn.edu/~ccb/ppdb/
- Where to use:
  - Query-Data matching

NLP Pipelines

- Apache UIMA
  - http://uima.apache.org/
- Open Advancement of Question Answering Systems (OAQA)
  - http://oaqa.github.io/
- OKBQA
  - http://www.okbqa.org/documentation https://github.com/okbqa
- Where to use:
  - Components integration

CS290D covers



- Open domain QA (TREC, 1999~2007)
- QA over linked data (~2007-)
- Recent Developments (~2013-)
  - April 29th

Questions?

Step 2: Template instantiation
Entity identification and predicate detection

- 1. For resources and classes, a generic approach to entity detection is applied:
  - Identify synonyms of the label using WordNet.
  - Retrieve entities with a label similar to the slot label based on string similarities (trigram, Levenshtein and substring similarity).
- 2. For property labels, the label is additionally compared to natural language expressions stored in the BOA pattern library.
- 3. The highest ranking entities are returned as candidates for filling the query slots.

Step 3: Query ranking and selection

- 1. Every entity receives a score considering string similarity and prominence.

- 2. The score of a query is then computed as the average of the scores of the entities used to fill its slots.

- 3. In addition, type checks are performed:
  - For all triples **?x rdf:type <class>,** all query triples **?x p e** and **e p ?x** are checked w.r.t. whether domain/range of p is consistent with <class>.

- 4. Of the remaining queries, the one with highest score that returns a result is chosen.