Semantic Parsing for Question Answering

Stefano Melacci

Nantes ML Meetup

November 7, 2016

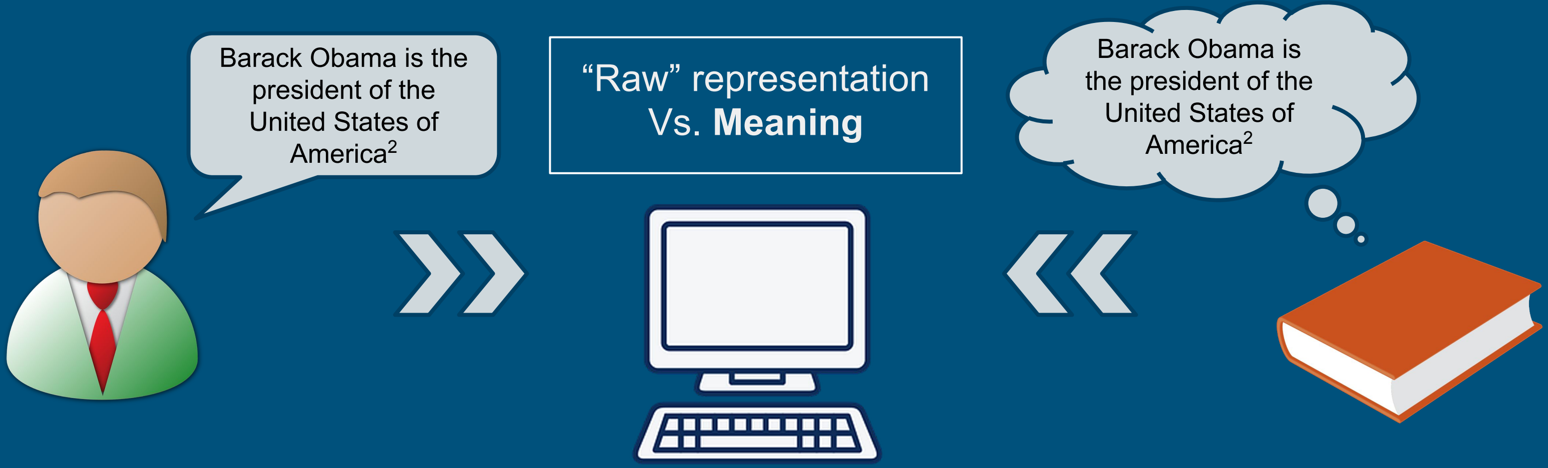
Overview

*I will introduce **Semantic Parsing for Question Answering** in a generic ("customized") way + I will explore **two existing approaches***

- (1) Semantic Parsing
- (2) Question Answering
  - Knowledge Bases (KBs)
- (1+2) Semantic Parsing for Question Answering with Structured KBs
  - Learning a Semantic Parser from Examples
- (Approach 1) Compositional Semantics
- (Approach 2) Query Graph Generation
- Discussion

Semantic Parsing

"Semantic parsing is the process of mapping a natural-language sentence into a formal representation of its meaning"[1]



1. http://www.cs.utexas.edu/~ml/publications/area/77/learning_for_semantic_parsing
2. Today is November 7, 2016… tomorrow is the Election Day

Semantic Parsing: Why?

- Multiple Domains
- Different Languages
- ...

Meaning

- Context Modeling
- ...

High-Level Understanding

Applications

- Question Answering
- Doc Classification
- Conversational Systems
- ….

Representing the Meaning of a Sentence

“Semantic parsing is the process of mapping a natural-language sentence into a formal representation of its meaning”

- **Natural Language**
  - Ambiguous, redundant, ...
- **Formal language**
  - Unambiguous, concise, …
  - We have to define the symbols, the rules, and all the features of the formal language we want to use to model the meaning of a sentence
  - Several possible choices

Example: First-Order Logic

"A man is driving a car"

$$\exists\, x,y.\ man(x) \land driving(x,y) \land car(y)$$

… that's not enough for building a Semantic Parser!

- Here we have an **uninterpreted** logical form
- Predicate symbols (*man*, *driving*, *car*) do NOT have **meaning** in themselves

Semantic Parsers

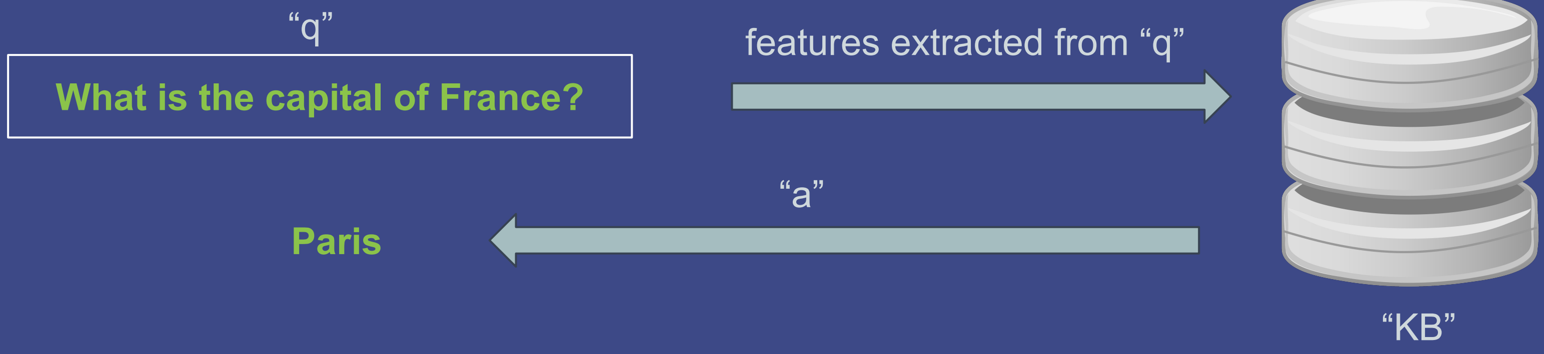
A Semantic Parser includes three main components[1]

1. **Formal Language**
   a. Recall the previous example of FOL
2. **Ontology**
   a. Predicate symbols are from a given ontology
   b. That's where they get their meaning!
3. **Inferential Mechanism**
   a. It is what takes a problem represented in the formal language and the ontology and performs the target task (e.g. **Question Answering on a given Knowledge Base**, etc.)

1. Beltagy, Katrin, Mooney. Semantic parsing using distributional semantics and probabilistic logic. ACL 2014

Question Answering

- We are given an input question *“q”*
- We also have the use of a Knowledge Base *“KB”* (source of knowledge)
  - Collection of documents (Unstructured KB)
  - Structured KB
- **Goal**: provide an answer *“a”* using the information in *“KB”*



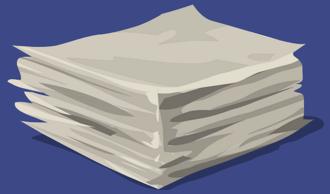
Question Types (Most Common)

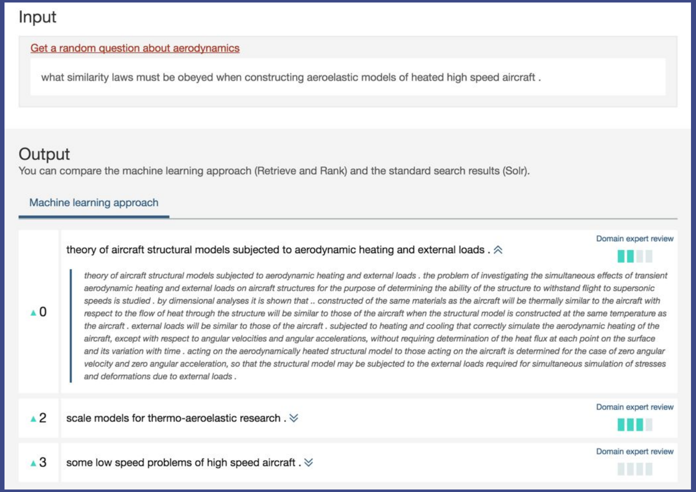
- What are the types of question that are commonly considered?
  - **Factoid:** *What is the capital of France?*
  - **Yes/No:** *Does a penguin fly?*
  - **List:** *What are the presidents of the U.S. of the last two decades?*
  - **Definition:** *What is a cat?*
  - **How:** *How do birds fly?*
  - **Why:** *Why does a rainbow form?*

Question Answering + Unstructured KB

The KB is a collection of documents, the Web, any batch of data that is not deeply structured accordingly to a given ontology, ...

- Search engine-like procedure
- "Retrieve and rank" (see IBM Watson)
- The answer is a **text passage**

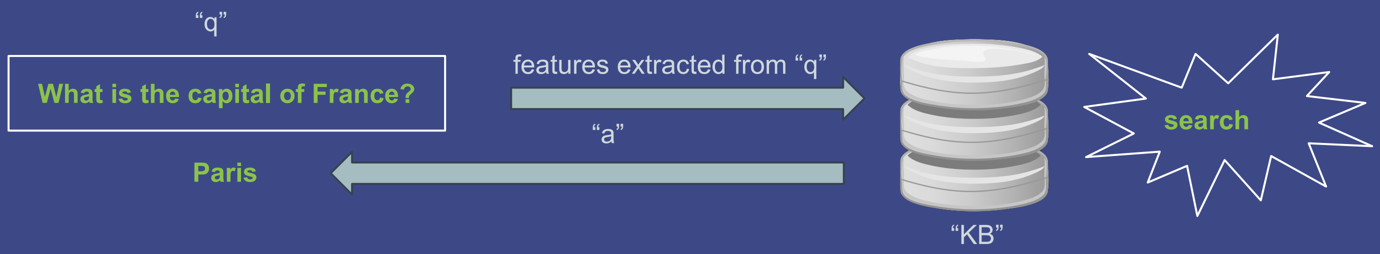




Snapshot taken from: IBM Watson, Retrieve and Rank Demo, http://retrieve-and-rank-demo.mybluemix.net/rnr-demo/dist/#/

Question Answering + Unstructured KB

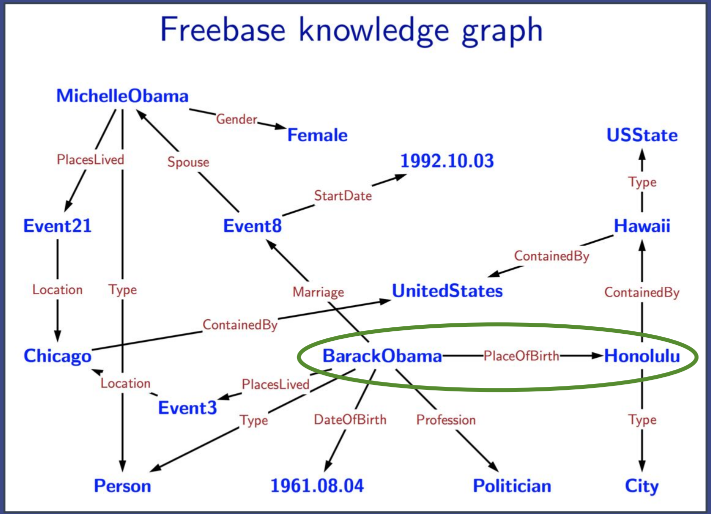
- Features
  - Expected answer type (*Person, Date, Location, Value, …*)
  - Keywords
  - ...
- Search
  - Identify candidate elements of the knowledge base by matching keywords, filtering, ...
  - Find a “nearby” answer of the expected type



Structured KB: The Freebase Example

Freebase is a good example of triple-based Structured KB (*Knowledge Graph*)

- It's a graph: Node-Edge-Node connections
  - **"Triples"** or **Binary Relations** between **Entities**
- Stats
  - > 56,000,000 nodes
  - > 3,000,000,000 triples
  - Several edge-types
- Collaborative project
  - Metaweb, then Google, then public…
  - ...then shut down (May 2, 2016)



Picture taken from: Percy Liang, SEMPRE: Semantic Parsing with Execution. https://web.stanford.edu/class/cs224u/materials/cs224u-sempre-slides.pdf

PlaceOfBirth(BarackObama, Honolulu)

Question Answering + Structured KB

The KB organizes factual knowledge in a structured manner

- Activate and navigate portions of the graph
  - Exploiting information on the graph structure
- The answer is an **entity of the KB**
  - A node of the graph

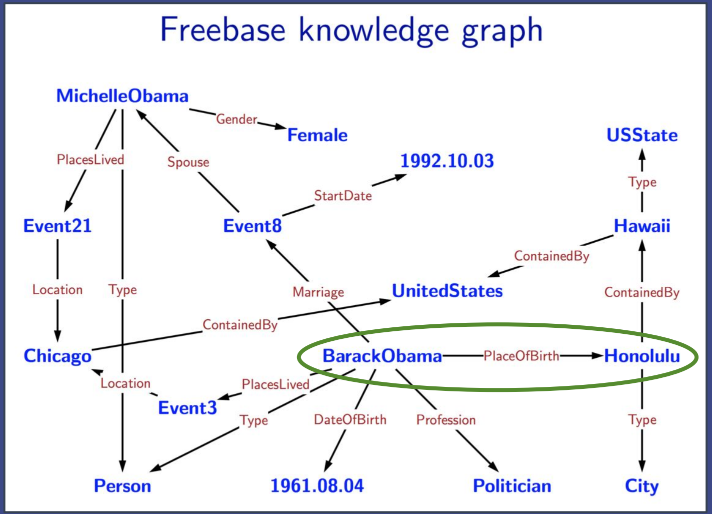
“q”

Where was Obama born?

features extracted from “q”

“a”

Honolulu



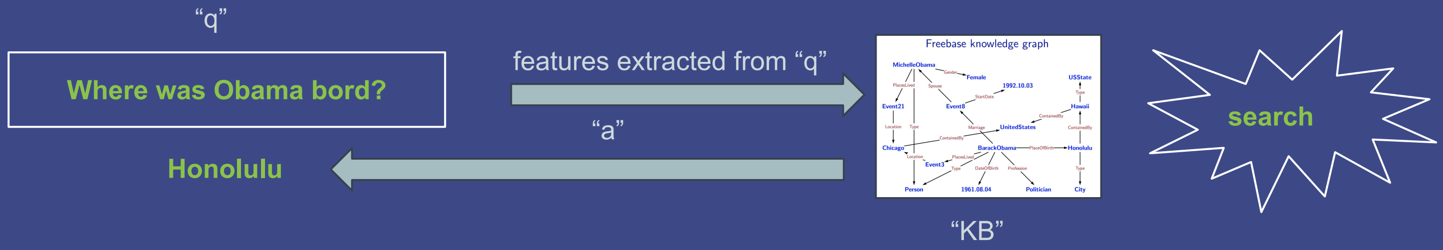
PlaceOfBirth(BarackObama, Honolulu)

Question Answering + Structured KB

What are the features/search procedure?

- Features: “*formal representation of the meaning of the question*”
- Search: “navigate” the graph given the representation of the meaning

Semantic Parsing...for QA



Semantic Parsing for QA with (Struct.) KB

Where was Obama born?

Alignment to KB

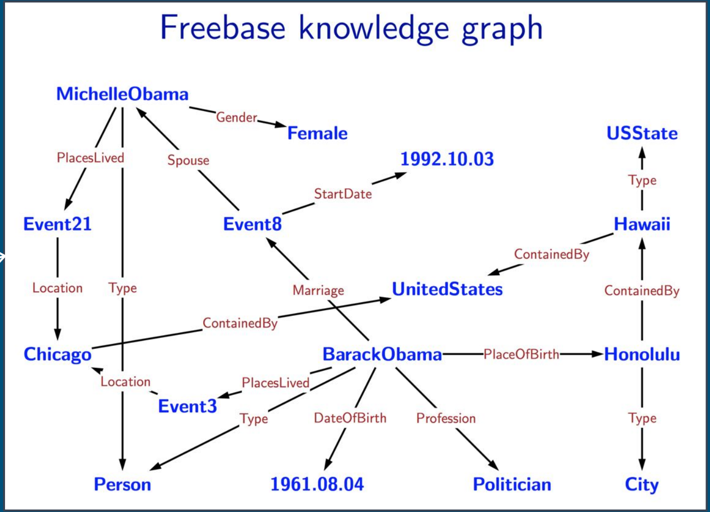
KB Relations and Entities
(multiple hypotheses)

Composition

Logical Form

("Execute")

Honolulu



Alignment (Multiple Hypotheses)

What is the capital of the state in which Hollywood is located?

question type:

Factoid

It returns “x”

KB entity (node)

“capital”

KB relation (edge)

“capitalOf(x,y)”

KB entity (node)

“state”

KB entity (node)

“status”

KB relation (edge)

“statesSomething(x,y)”

KB entity (node)

“Hollywood”

is-a(*Hollywood*, ***city***)

KB entity (node)

“Hollywood”

is-a(*Hollywood*, ***animal***)

KB relation (edge)

“locatedIn(x,y)”

KB relation (edge)

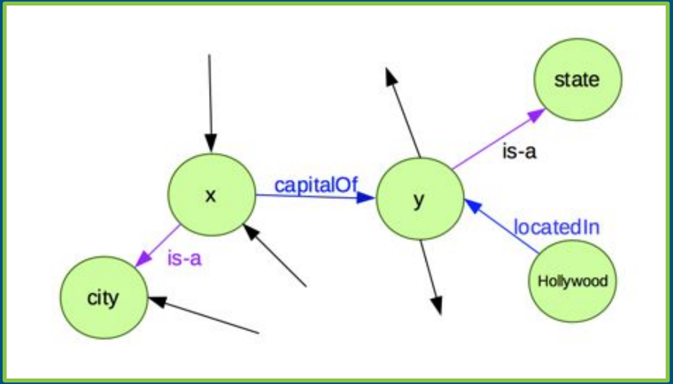
“locate(x,y)”

KB relation (edge)

“is-a(x,y)”

Composition (Multiple Results)

- Given multiple hypotheses of alignment, we have to compose them
  - Associate **entities** to the arguments of the **relations**
  - Connect arguments of multiple relations
  - Introduce new relations (Bridging)
  - Exploit the question/answer type



What is the capital of the state in which Hollywood is located?

We are building a subgraph of the KB + free variables!

Logical Form

- Sometimes we also need to introduce aggregation operators
  - *argmax, argmin, …*
- Finally, we can represent the composed meaning of the sentence with a selected formalism, generating a **Logical Form**

What is the most populous city in California?

$$\texttt{argmax}(\lambda x.\texttt{city}(x) \wedge \texttt{loc}(x, \texttt{CA}), \lambda x.\texttt{population}(x))$$

“Executing” the Logical Form returns the answer - what is associated to the free variable *x* in the KB

Example from: Liang, Jordan, Klein - Learning Dependency-Based Compositional Semantics - Computational Linguistics 2013

Semantic Parsing for QA with (Struct.) KB

Where was Obama born?

↓

Alignment to KB

↓

KB Relations and Entities
(multiple hypotheses)

↓

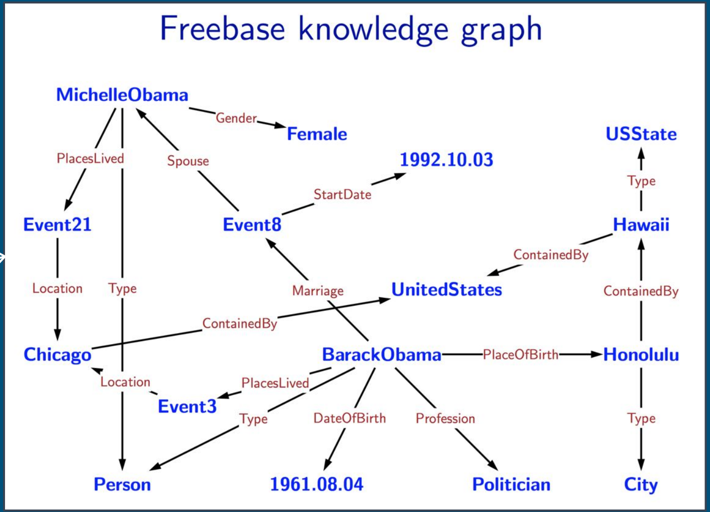
Composition

↓

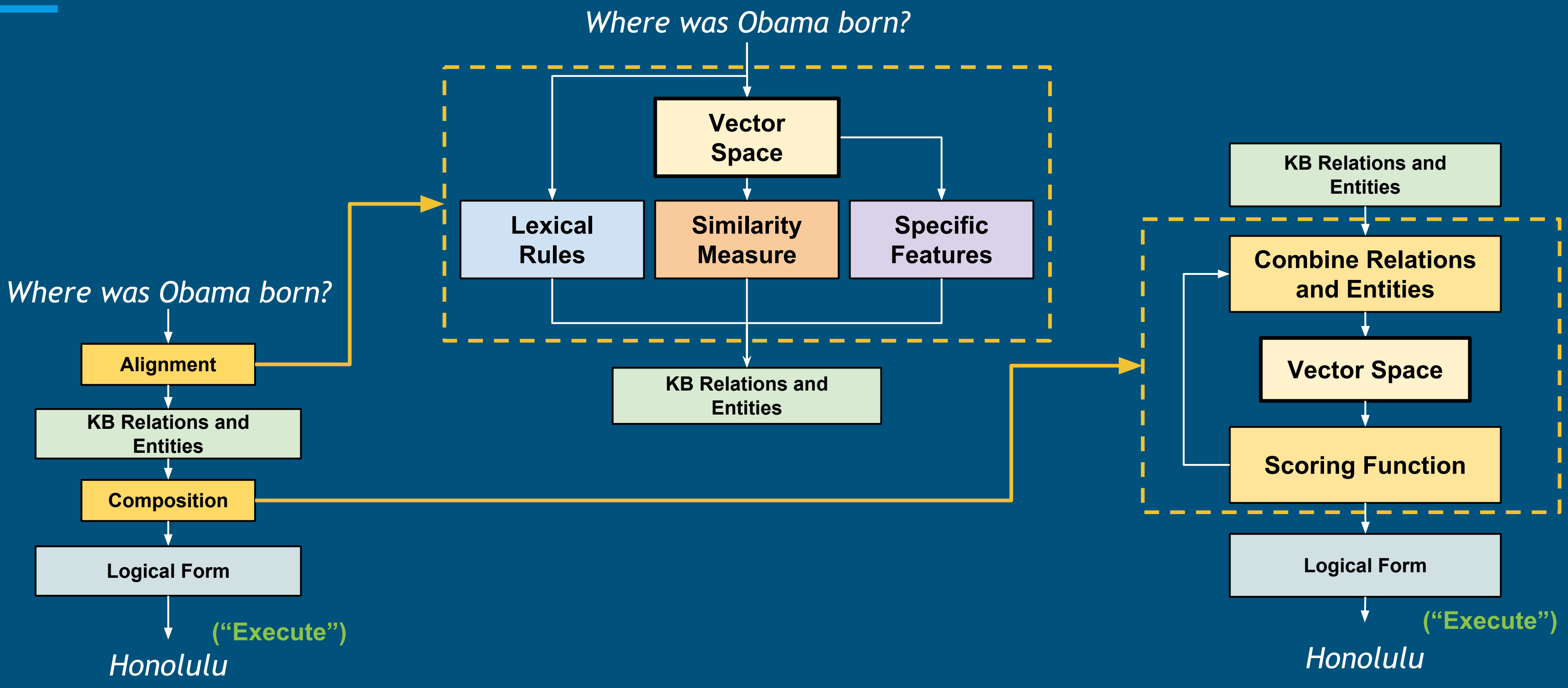
Logical Form

↓ **("Execute")**

Honolulu

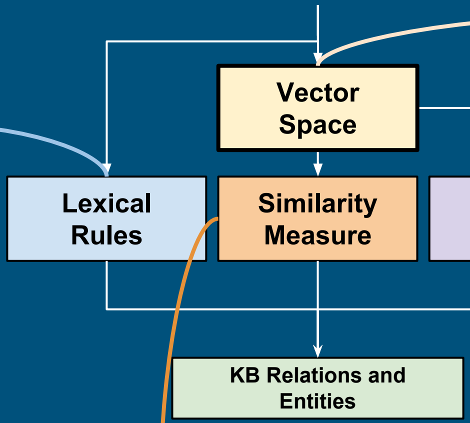


Semantic Parsing for QA with (Struct.) KB



Different ways of Aligning text to KB

Where was Obama born?



- *One or more rules for each KB relation and KB entity*
- Rules involve the words of the question
- They are also based on the POS tagging

Compare the input utterance to all the KB relations and entities

- Dot product
- Cosine similarity
- ...

A feature (function) to predict each KB rule and KB entity

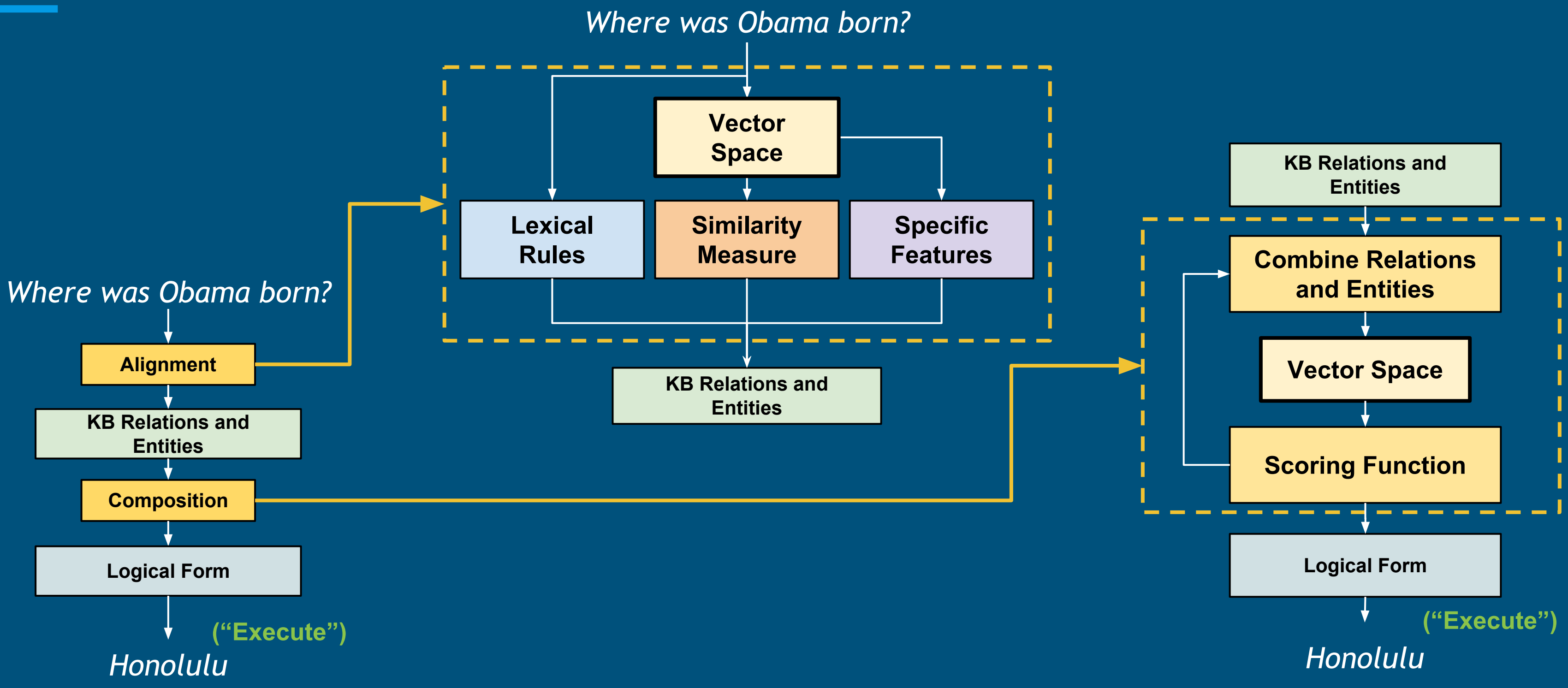
Several possible choices for encoding the input question

- Multiple vocabularies (for input tokens and for KB relation names or entity names)
- Letter-n-grams
- Word-n-grams

Then

- BoWs, RNNs, CNNs, …

Semantic Parsing for QA with (Struct.) KB



Details on the Composition stage

Define composition rules, to generate a set of candidates "*meanings*" (Logical Forms)

- Rules are applied in function of the KB contents

Example

- **Join**
  From: drinks(x,y), Beer
  To: drinks(x, Beer)
- **Intersect**
  From: friendOf (x,y), bornIn(z, Paris)
  To: friendOf(x,y) ^ bornIn(y, Paris)
- **Bridge**, **Aggregate**, ...

The candidate "*meanings*" (Logical Forms) and the *question* are projected into a vector space

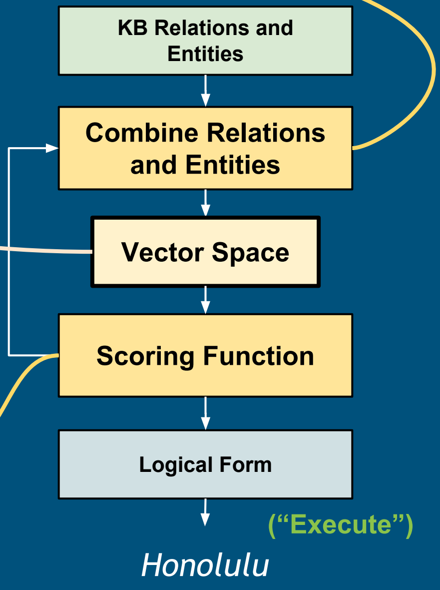
- Ad-hoc features
- Learned Embeddings

The scoring function ranks the pairs (*question*, "*meaning*" / Logical Form)

Huge number of candidate "meanings"

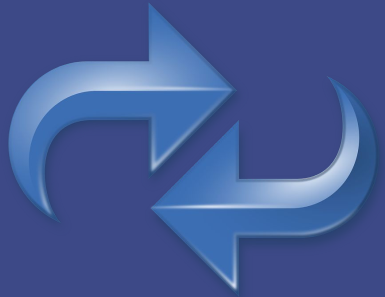
- **Beam Search** + **Scoring Function** while composing, in order to prune unpromising partial results

Best-ranked logical form is **executed**, we get the *answer* to the input question



Learning a Semantic Parser from Examples

- What supervision signal can we use?
  - (1) Pairs of **question + logical form ("meaning")**
    - Expensive but precise
    - Logical forms are observed variables
  - (2) Pairs of **question + answer**
    - Cheaper (weaker)
    - Logical forms are latent variables



The research community essentially went from (1) to (2), but we are also recently observing some approaches that follow the opposite tendency (seq-to-seq)

Learning from QA pairs: Criterion

We can learn the parameters (***θ***) of a Semantic Parser model by maximizing the probability of returning the right answer (***y***) for the estimated Logical Form (***z***) of the question (***x***), given a structured KB (***w***)

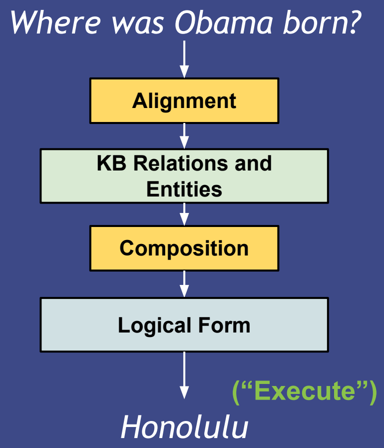
Objective:

$$\max_{\theta} \sum_{z} \underbrace{p(y \mid z, w)}_{\textbf{Interpretation}} \; \underbrace{p(z \mid x, \theta)}_{\textbf{Semantic parsing}}$$

Liang, Jordan, Klein - Learning Dependency-Based Compositional Semantics - Computational Linguistics 2013

Learning from QA pairs: What do we learn?

- The supervision signal is back-propagated
  - Some methods do not learn the **alignment** stage (*Lexical Rules*)
  - Other works decouple the learning process of the **alignment** and **composition** stages (assumptions!)
- Several assumptions on the type of questions that can be handled
  - *"Simple Questions"*
  - The composition stage can become extremely trivial



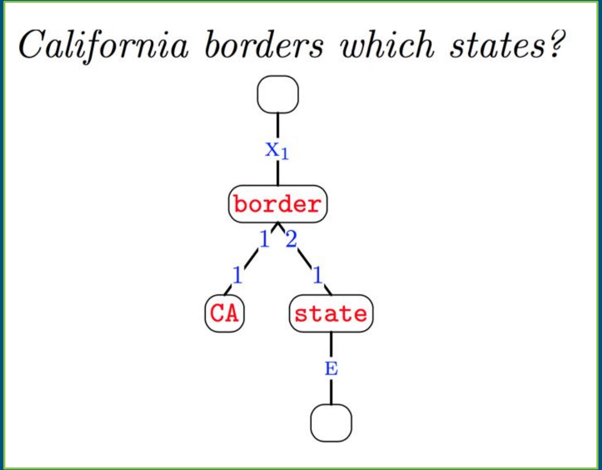
Objective:

$$\max_{\theta} \sum_{z} \underbrace{p(y \mid z, w)}_{\text{Interpretation}} \underbrace{p(z \mid x, \theta)}_{\text{Semantic parsing}}$$

Approach 1. Compositional Semantics (DCS)

Liang, Jordan, Klein. Learning dependency-based compositional semantics. ACL 2011 / Computational Linguistics 2013

- Logical forms are represented by a "new semantic representation"
  - Dependency-based Compositional Semantics (**DCS**)
  - The logical forms (in this framework) can be represented as special trees
    - Closer to the syntax (with some exceptions)
    - Easier to "visualize"



Approach 1. Knowledge Base

- We are in the setting described so far
  - But the KB is not Freebase!

- The authors focus on a database of "geography"-related facts
  - Each table models a **predicate**
  - It does not look like a graph…
  - *... but we could represent it as a graph*

city

San Francisco
Chicago
Boston
...

loc

Mount Shasta	California
San Francisco	California
Boston	Massachusetts
...	...

state

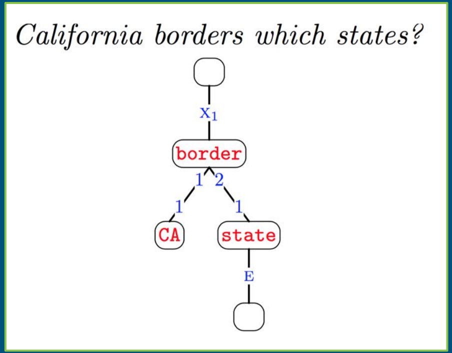
Alabama
Alaska
Arizona
...

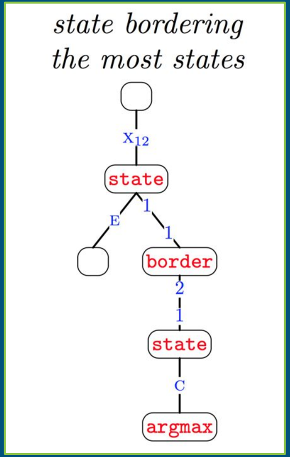
population

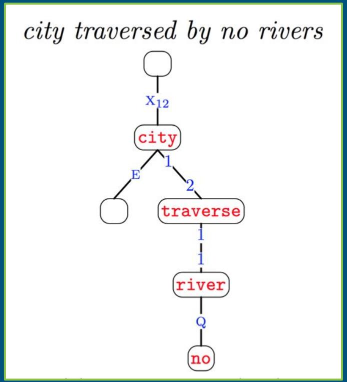
Los Angeles	3.8 million
San Francisco	805,000
Boston	617,000
...	...

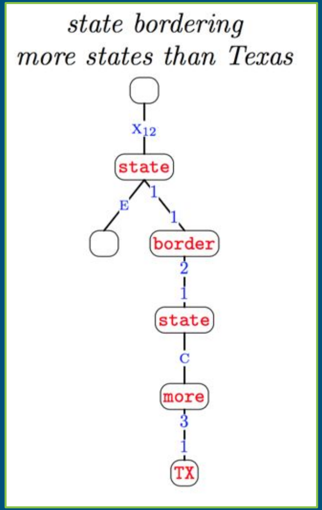
Approach 1. Representation of "Meaning"

- Example of questions and Logical forms represented as DCS trees







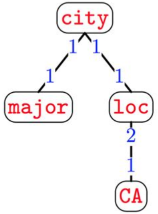


Approach 1. Trees and Logical Forms

- Don't be confused by the trees, we are still talking about logical forms!
  - The DCS tree is just a convenient way of representing the logical form

Example: *major city in California*

$$z = \langle \texttt{city}; \tfrac{1}{1}{:}\langle \texttt{major} \rangle \,; \tfrac{1}{1}{:}\langle \texttt{loc}; \tfrac{2}{1}{:}\langle \texttt{CA} \rangle \rangle \rangle$$



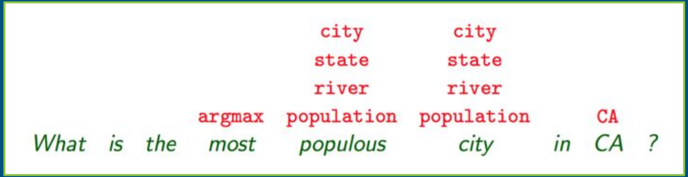
$$\lambda c\, \exists m\, \exists \ell\, \exists s\, .\ \texttt{city}(c) \wedge \texttt{major}(m) \wedge \texttt{loc}(\ell) \wedge \texttt{CA}(s) \wedge c_1 = m_1 \wedge c_1 = \ell_1 \wedge \ell_2 = s_1$$

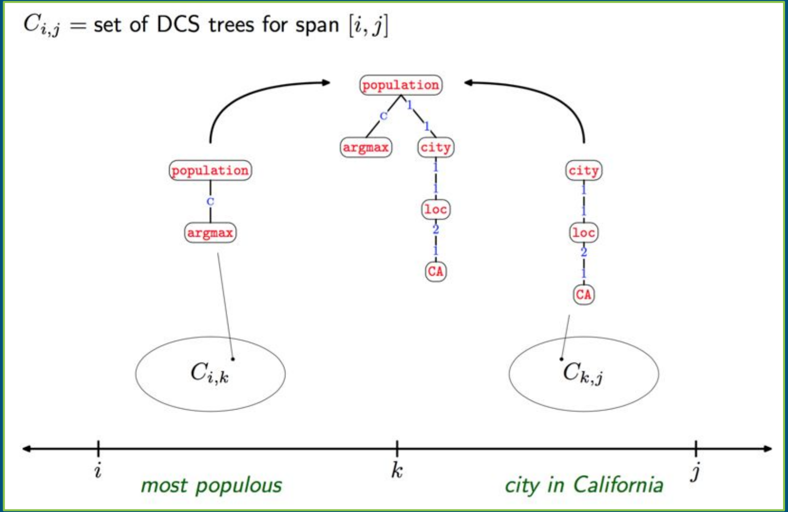
(a) DCS tree (b) Lambda calculus formula

(c) Denotation: $[\![z]\!]_w = \{\texttt{SF}, \texttt{LA}, \ldots\}$

Approach 1. Alignment and Composition

- Alignment
  - Lexical triggers for entities and relations
  - They can be task-specific
- Composition
  - Recursive DCS tree building
  - Large number of results...
  - ...Scoring (log-linear) + Beam Search

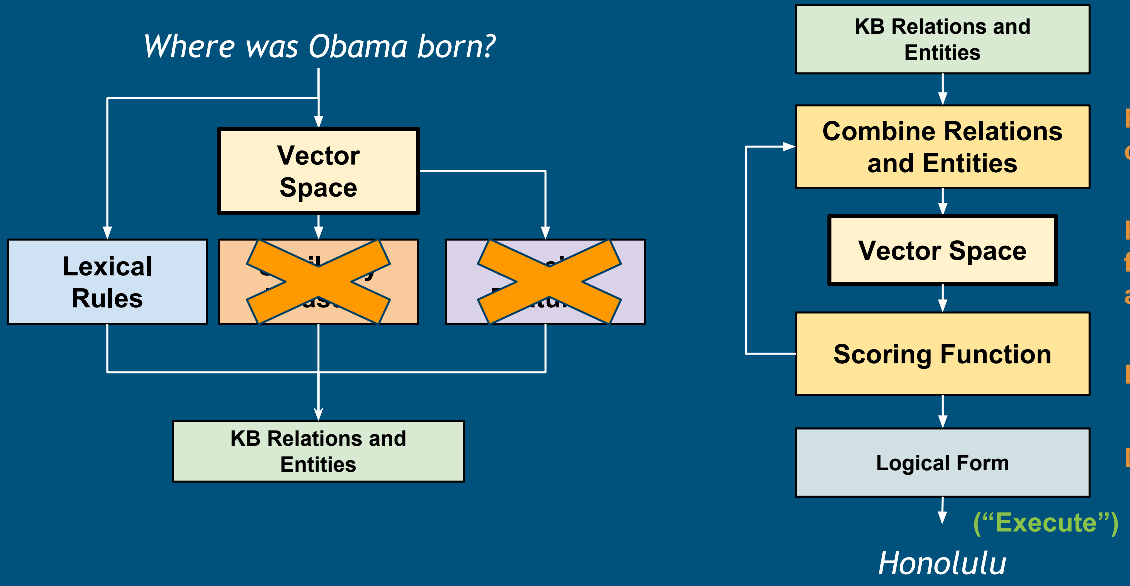




Pictures taken from: Percy Liang's slides presenting this approach

Approach 1. Global View

* Indicator feature templates: total number of predicates, statistics on words triggering predicates, ...



Lexical Triggers (fixed)

Example: the word "city" activates the predicate city(x)

Recursive DCS tree construction

Hand-designed features about DCS tree and the question*

Log-linear model

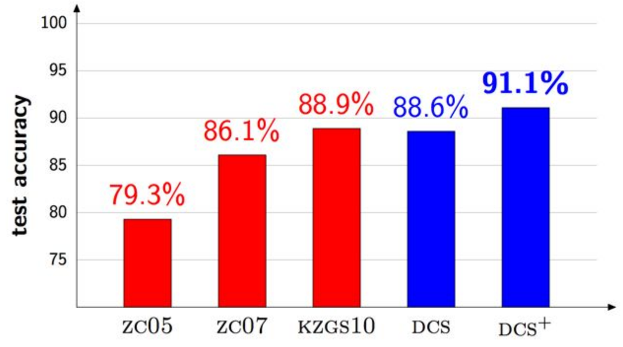
DCS Tree

Approach 1. Experiments

- Experimental Results
  - With specific Lexical Triggers
  - With “more generic” triggers

On GEO, 600 training examples, 280 test examples

System	Description	Lexicon	Logical forms
ZC05	CCG [Zettlemoyer & Collins, 2005]	✗ ✗	✓
ZC07	relaxed CCG [Zettlemoyer & Collins, 2007]	✗ ✗	✓
KZGS10	CCG w/unification [Kwiatkowski et al., 2010]	✗ ✗	✓
DCS	our system	✓ ✗	✗
DCS$^+$	our system	✓ ✓	✗



Graph traken from: https://cs.stanford.edu/~pliang/papers/dcs-acl2011-talk.pdf

Approach 2. Query Graph Generation

`Yih, Chang, He, Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. Int. Conf. of the Association for Computational Linguistics 2015`

- Logical forms are represented as small graphs
    - Query Graph
- Deep Neural Networks to align relations
- *"Simple Questions"*

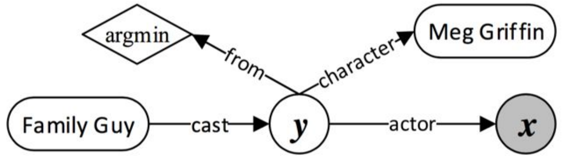
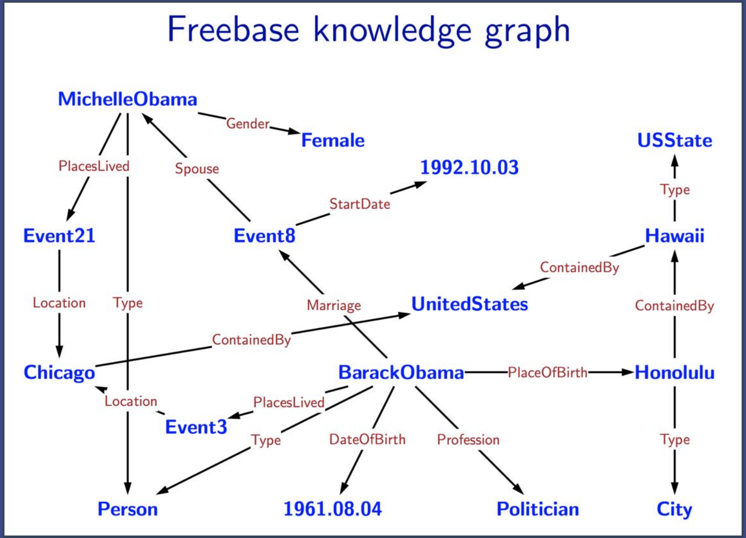


Figure 2: Query graph that represents the question "Who first voiced Meg on Family Guy?"

Approach 2. Knowledge Base

- **Freebase**



Approach 2. Representation of “Meaning”

- The structure of the Query Graph is due to the structure of Freebase
  - **y** = **EventX** nodes

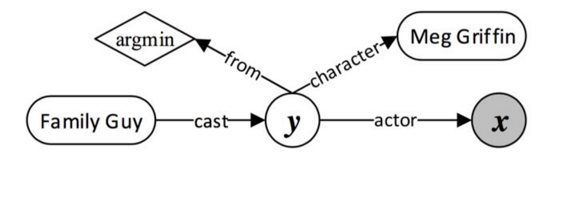
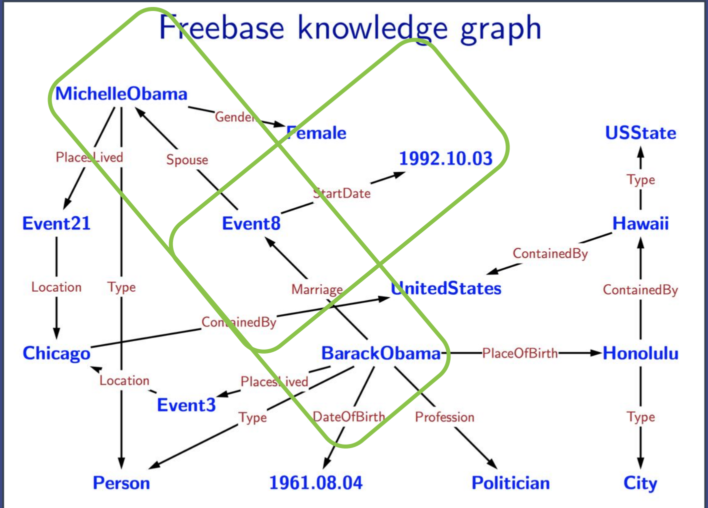


Figure 2: Query graph that represents the question “Who first voiced Meg on Family Guy?”

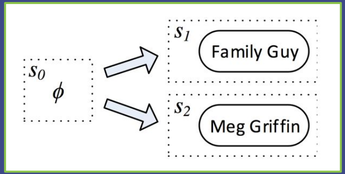


Approach 2. Alignment and Composition

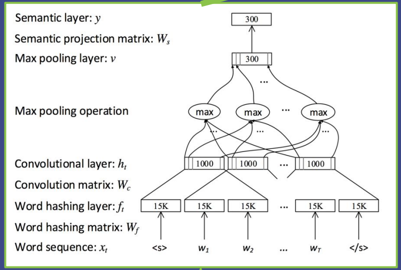
- Iterative procedure to build the **Query Graph**

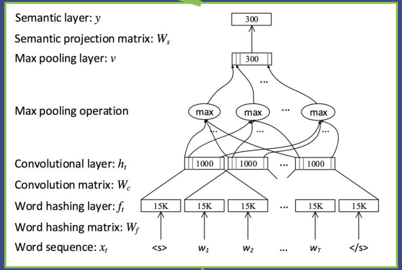
Similarity Measure

Who first voiced Meg in Family Guy?



1. Align the main entity





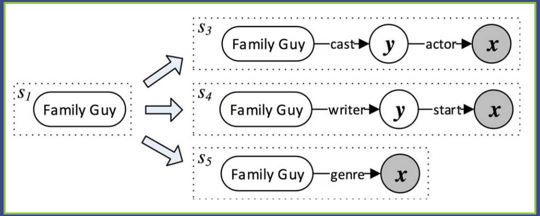
Who first voiced Meg in <e>?

cast-actor

...

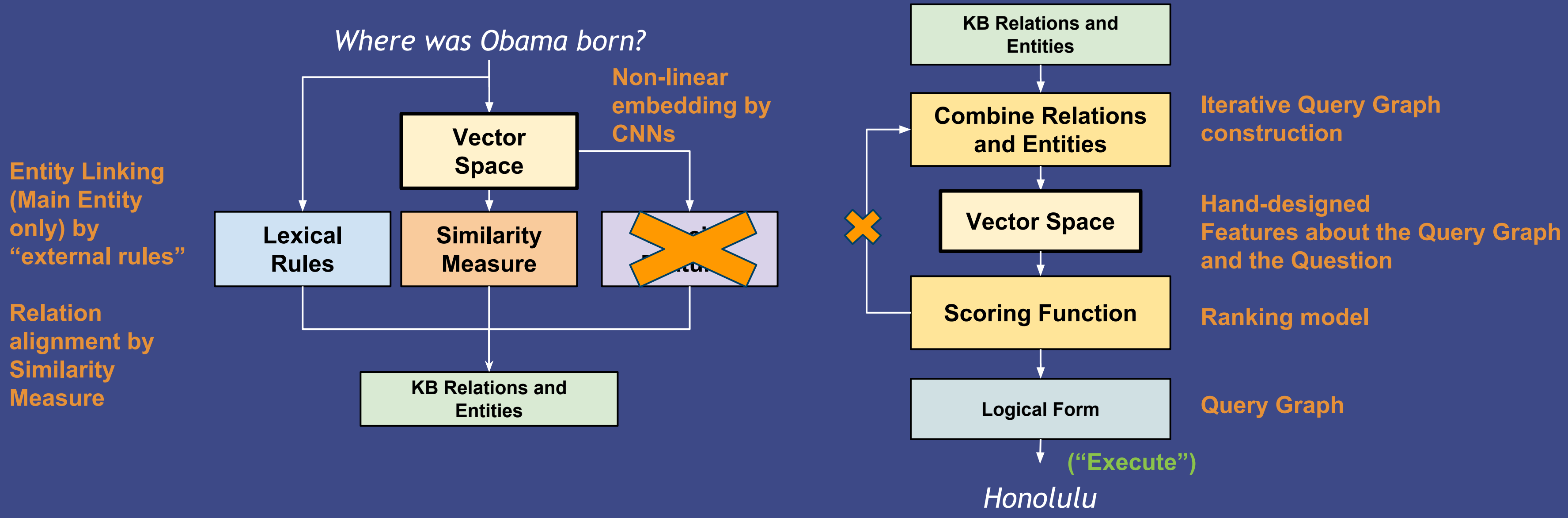
...

2. Align relations



3. Generate query graphs using the aligned entity/relations (+ add other fancy stuff)

Approach 2. Global View



Approach 2. Experiments

- WebQuestions dataset
  - 6K QA pairs
  - “Simple Questions”
    - Single fact involved

Method	Prec.	Rec.	F_1
(Berant et al., 2013)	48.0	41.3	35.7
(Bordes et al., 2014b)	-	-	29.7
(Yao and Van Durme, 2014)	-	-	33.0
(Berant and Liang, 2014)	40.5	46.6	39.9
(Bao et al., 2014)	-	-	37.5
(Bordes et al., 2014a)	-	-	39.2
(Yang et al., 2014)	-	-	41.3
(Wang et al., 2014)	-	-	45.3
Our approach – STAGG	52.8	60.7	**52.5**

- They authors of Approach 2 also need the relation labels to train the CNNs!
  - They artificially create them

Discussion

Semantic Parsers for QA

- Explicit representation of the “meaning” of a question
- Structured Knowledge Bases
- Learnable using QA pairs

On the other hand...

- The KB is given, not “learned”
- The compositionality is really stressed only in small datasets

Thank you!