Neural Symbolic Machines:

Learning Semantic Parsers on Freebase with Weak Supervision

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, Ni Lao, ACL 2017

Presenter: Qiao Han (hanqiao@seas)
02/26/20

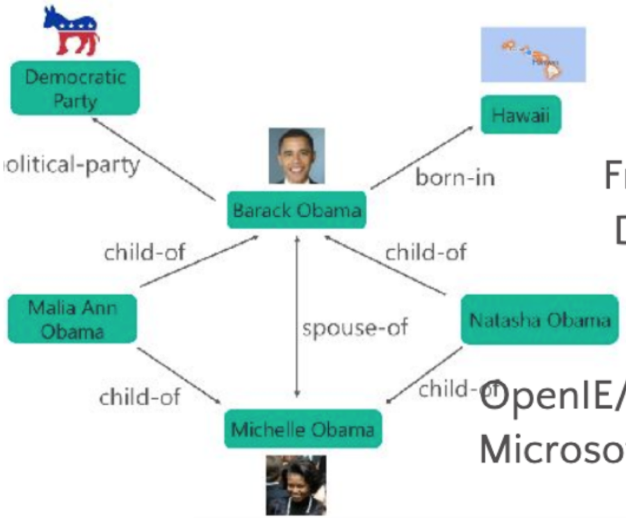
Penn Engineering

Contents:

- **Problem and Motivation**
- Previous SOTA
- Neural Symbolic Machine
  - Framework
  - Neural Programmer
  - Symbolic Computer
- Results and Analysis

Background — Knowledge Graph

- Let E denote a set of entities
- Let P denote a set of relations
- A knowledge base K is a set of assertions or triples (e1, p, e2) \in E x P x E
- Example:



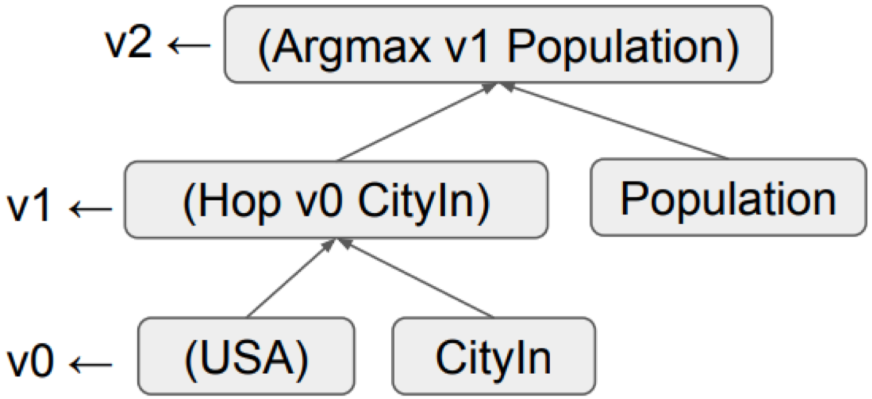
Freebase
DBpedia
YAGO
NELL
OpenIE/ReVerb
Microsoft Satori

[Liang, et al 2017]

Penn Engineering

Background — Semantic Parsing

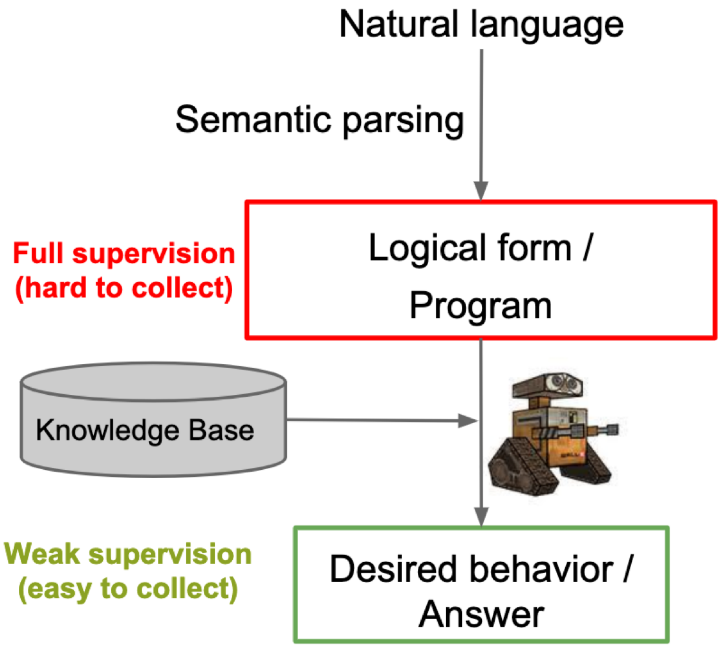
- Given a knowledge base K, and a question q = (w1, w2, …, wk)
- Produce a program or logical form z such that
- When (z) executed against K generates the right answer y



ang, et al 2017]

Problem & Motivation

- QA with weak supervision



[Liang, et al 2017]

Problem & Motivation

- QA with large KB

Large-scale Knowledge Base

Properties of Hundreds of millions of entities

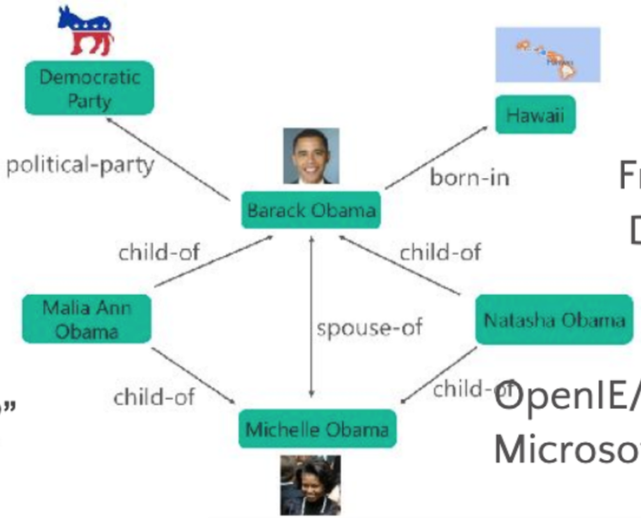
Plus relations among them

E.g. Freebase, 26k predicates, 200M entities, 3B triples

Question Answering

"What are the names of Obama's daughters?"

$\lambda x.\text{parent}(\text{Obama},x) \cap \text{gender}(x,\text{Female})$

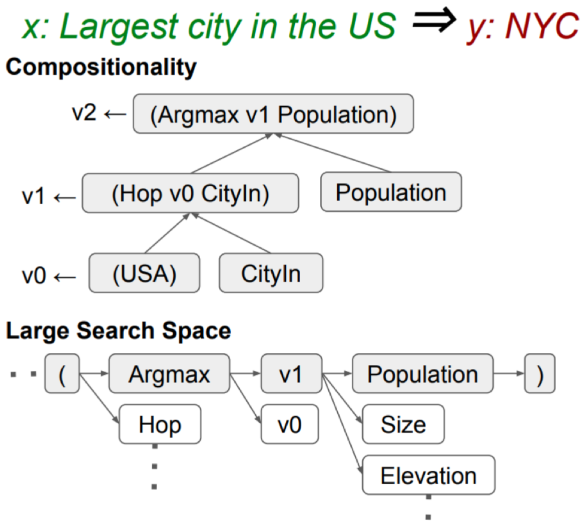


Freebase
DBpedia
YAGO
NELL
OpenIE/ReVerb
Microsoft Satori

[Liang, et al 2017]

Problem & Motivation

- Key challenges:
  - Compositionality
    - the semantics of a question may involve multiple predicates and entities
  - Large search space
    - Some freebase entities have >160k immediate neighbors
    - 26k predicates in Freebase



[Liang, et al 2017]

Contents:

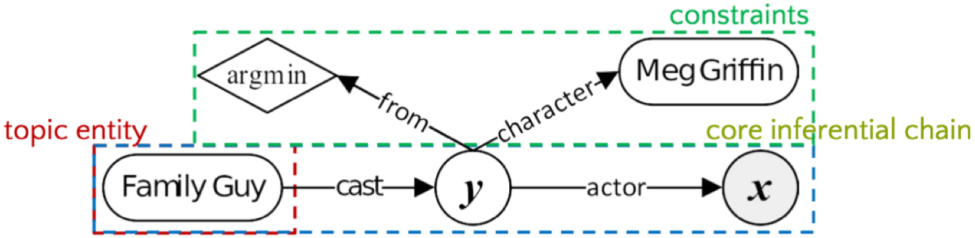
- Problem and Motivation
- **Previous SOTA**
- Neural Symbolic Machine
  - Framework
  - Neural Programmer
  - Symbolic Computer
- Results and Analysis

Previous SoTA — STAGG

- Staged Query Graph Generation (STAGG)

Who first voiced Meg on Family Guy?

$\lambda x. \exists y. \mathrm{cast}(\mathrm{FamilyGuy}, y) \wedge \mathrm{actor}(y, x) \wedge \mathrm{character}(y, \mathrm{MegGriffin})$

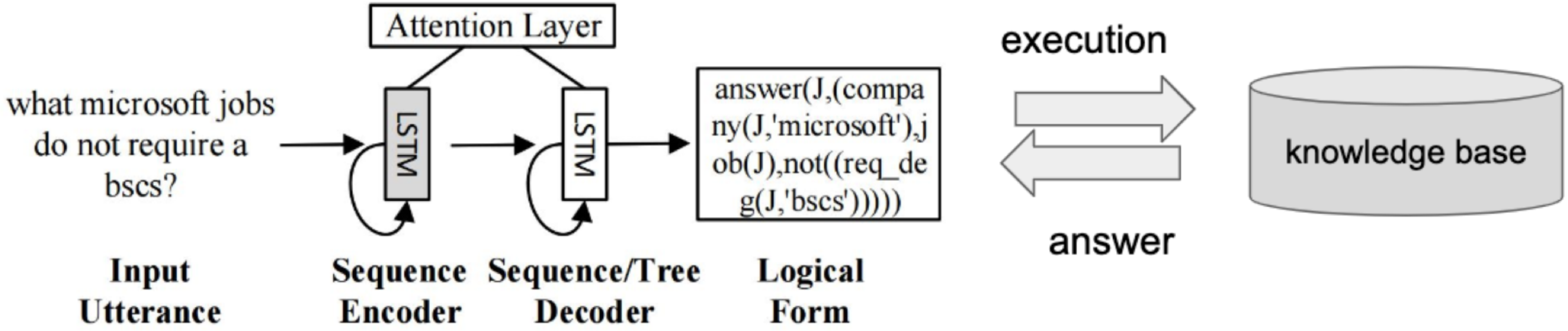


- Constraints of STAGG:
    - hand crafted rules and feature engineering

[Yih, et al 2016]

Previous SoTA — Pure Seq2Seq

- Constraints of Pure Seq2Seq in Semantic Parsing:
  - Requires strong/full supervision (e.g. JOBS, GEO)
  - Lack of integration/interaction with executer
  - Unable to search over exponentially large hypothesis



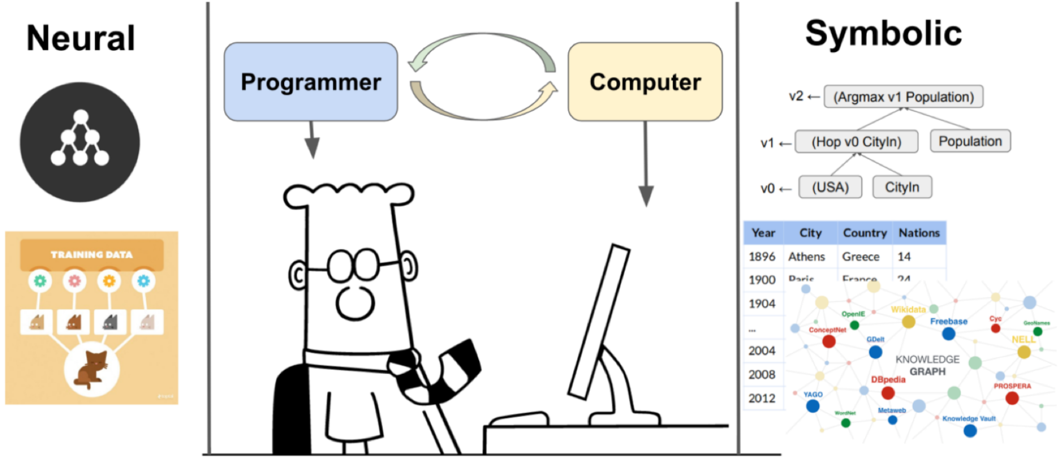
[Yih, et al 2016]

Contents:

- Problem and Motivation
- Previous SOTA
- **Neural Symbolic Machine**
  - **Framework**
  - Neural Programmer
  - Symbolic Computer
- Results and Analysis

Framework:Programmer-Computer

- Weak supervision + Large KB:



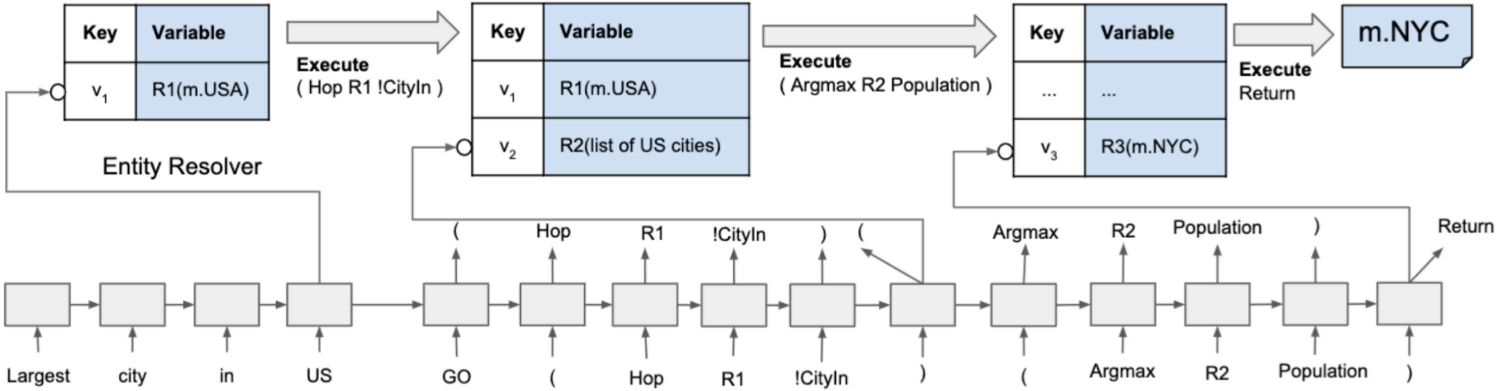
[Liang, et al 2017]

Contents:

- Problem and Motivation
- Previous SOTA
- **Neural Symbolic Machine**
  - Framework
  - **Neural Programmer**
  - Symbolic Computer
- Results and Analysis

Neural Symbolic Machine — Programmer

- Modifications to seq2seq model
  - Add a key-variable memory for compositionality
  - Memory is ‘symbolic’



[Liang, et al 2017]

Penn Engineering

Program — High-level Uniform Syntax

- A program is a list of **expressions** (c1, c2, …, cl)
- An expression c is either a special token "return" or a list (F A0, A1, …, Ak)
- F is one of the **functions** (e.g. HOP, ARGMAX, EQUAL, etc.)
- A_i can be either a **relation** p \in P, or a **variable** v
- A variable v is a special token representing a list of entities ([e1, e2, …, et])
- Examples: HOP v p; ARGMAX v p

Penn Engineering

Program — High-level Uniform Syntax

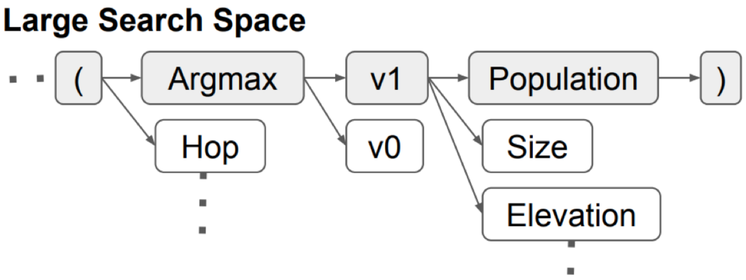
- Reduce search space:
  - syntax check
  - semantic check
  - predefined 22 functions (ARGMAX, FILTER_IN, FIRST, SUM, ...)

Contents:

- Problem and Motivation
- Previous SOTA
- **Neural Symbolic Machine**
  - Framework
  - Neural Programmer
  - **Symbolic Computer**
- Results and Analysis

Neural Symbolic Machine — Computer

- Non-differentiable, weak supervision training using REINFORCE
  - Cons:
    - sparse learning
    - cold initialization
    - spurious programs
    - large search space, slow execution



Neural Symbolic Machine — Computer

- Improvements:
  - sparse learning
    - Augmented REINFORCE with iterative ML
  - cold initialization
    - Large beam search, MAPO*
  - spurious programs*
  - large search space, slow execution
    - Code assistance*
    - Distributed Actor-Learner Architecture

*: improved upon later publications

Contents:

- Problem and Motivation
- Previous SOTA
- Neural Symbolic Machine
  - Framework
  - Neural Programmer
  - Symbolic Computer
- **Results and Analysis**

WebQuestions Dataset

- What The WebQuestions dataset is a question-answering benchmark created by Berant et al. (2013).
- WebQuestionsSP
  - Remove invalid QA pairs
  - Add logical form annotations
  - 3098 training and 1639 testing questions

Results and Analysis

- SoTA

Model	Avg. Prec.@1	Avg. Rec.@1	Avg. F1@1	Acc.@1
STAGG	67.3	73.1	66.8	58.8
NSM – our model	70.8	76.0	**69.0**	59.5
STAGG (full supervision)	70.9	80.3	71.7	63.9

- Augmented REINFORCE

Settings	Train Avg. F1@1	Valid Avg. F1@1
iterative ML only	68.6	60.1
REINFORCE only	55.1	47.8
Augmented REINFORCE	83.0	**67.2**

[Liang, et al 2017]

Conclusions, Shortcomings and Future Work

- Technical Takeaways
  - Computer Programmer Framework for Weak Supervision
  - Various Optimizations
    - Symbolic Semantic Encoding
    - Augmented REINFORCE
- Reasoning?
  - KB-based, but no context understanding, multi-hop inferences required
  - knowledge outside of given KB?
  - extraction vs. discovery?

Conclusions, Shortcomings and Future Work

- Shortcomings/Insufficiencies
  - Overstatement on weak-supervision
  - Unstable REINFORCE*
  - Not significant improvements above benchmarks
  - Overfitting
- Future work?
  - expand KG to include more schema
  - more data, stable results

Questions?