CS224N Final Projects

10/6/2014
The course staff and volunteers

Past Projects

- Past projects reports:
  - http://nlp.stanford.edu/courses/cs224n/

- List of current project ideas, project description, data **will** be available on the website
  - Sorry they are not already there

Sentence Vectors – Sam Bowman

- How does sentence embedding accuracy scale with sentence size and embedding size?

- Fixed length vector representations of sentences – generated from neural network models like recursive NNs or LSTMs – are fairly widely used in NLP (most recently in some groundbreaking translation work)

References

- Neural Machine Translation by Jointly Learning to Align and Translate
  - http://arxiv.org/pdf/1409.0473.pdf
- Dynamic pooling and unfolding RAE
- Contact Sam Bowman, sbowman@stanford.edu
- Skill in MATLAB helps

Chat bot with real NLP - Gabor

- Contact: Gabor Angeli, angeli@stanford.edu
- “I've always found it somewhat strange that there's little to no NLP research [anymore] on making chat bots. The result is that most seem to blindly parrot other user's input, or revert to inane general conversation.”

Chat bot with real NLP - Gabor

The project proposal is to combine a number of common NLP techniques and systems to create a more ""intelligent"" chat bot. For instance:

- Learn common conversation patters from subtitles / scripts
- Add in SemPre (https://github.com/percyliang/sempre) for freebase factoid Q/A
- Make a few cute custom grammars for SemPre
- Throw in a simple sentiment detector, and tailor comments appropriately.

Improving vector space models

Contact: Thang Luong lmthang@stanford.edu

Learning distributed representations for variable-length texts

Using global context in joint neural language model

lmthang@stanford.edu

Learning beyond word vector representations

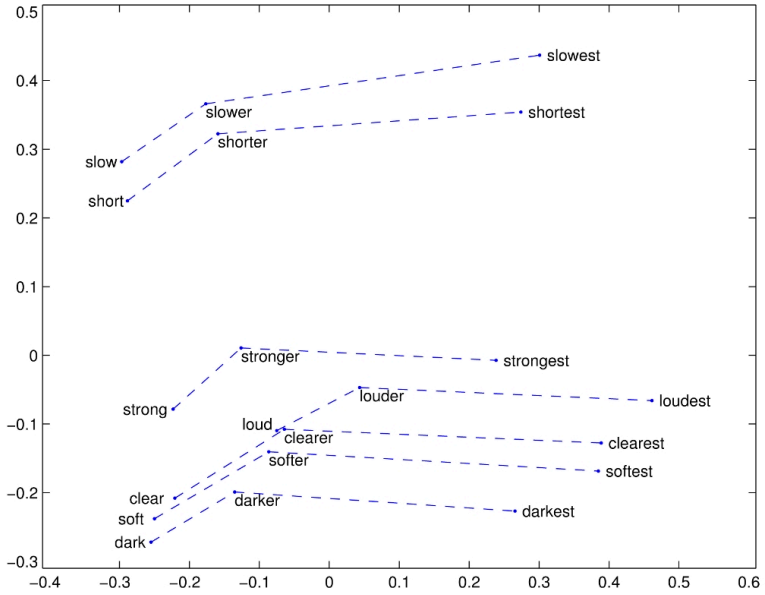
0. *frog*
1. frogs
2. toad
3. litoria
4. leptodactylidae
5. rana
6. lizard
7. eleutherodactylus

3. litoria

4. leptodactylidae

5. rana

7. eleutherodactylus

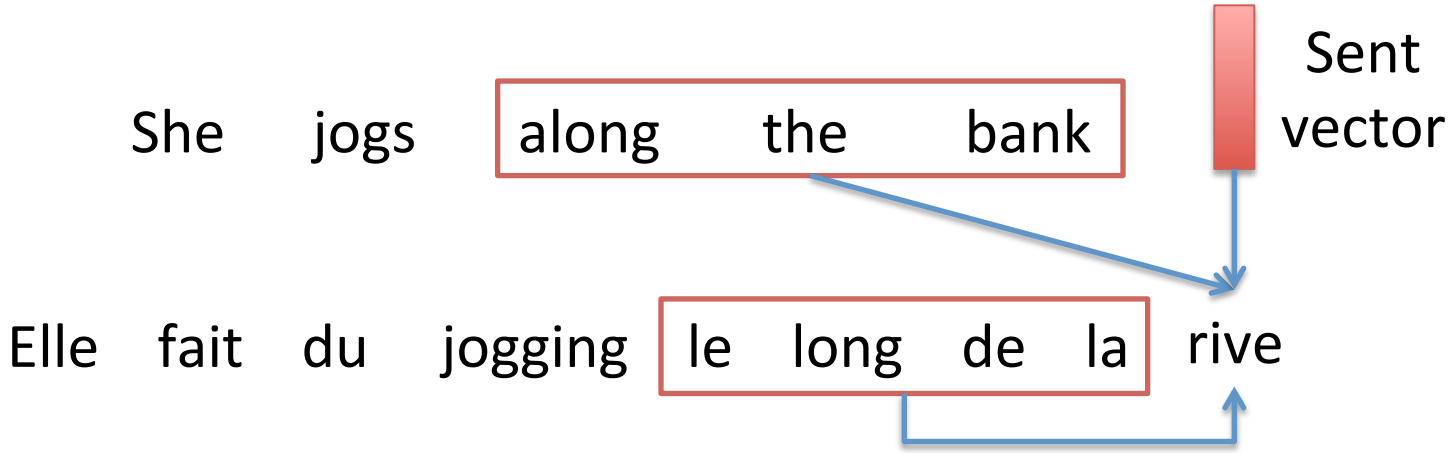


- Learn sentence/paragraph/ document vectors.
- Want to be fast at test time.
- Extend word2vec (C++)

Pictures from http://www-nlp.stanford.edu/projects/glove/

lmthang@stanford.edu

Joint neural language model for MT



- Joint LM: conditioned on the source sentence.
- Use global information: sentence vector
- Extend existing code in Python/Theano.

ML-based tokenizer

Christopher Manning manning@stanford.edu

The Stanford NLP tools use a hand-written deterministic FSM tokenizer. It's actually pretty good, but it can't consider word forms that are natural to humans

ML-based tokenizer

so it doesn't get right things like when two words are runtogether and nor can it decide when a parenthesis followed by a colon is intended punctuation (which sometimes happens): versus when it is a backwards frown face, which also sometimes happens.):

ML-based tokenizer

Christopher Manning manning@stanford.edu

Could a machine learning classifier work better?

It could be a sequence model done as a CRF like Chinese word segmentation, but could well do better using longer range context like a word-level language model or certain kinds of recurrent neural net. There has been some prior work on this using somewhat unsupervised means (search punkt).

Automatically improving dependency annotation

Natalia Silveira natalias@stanford.edu

Quality annotations are crucial for many NLP applications, but annotating large amounts of data consistently and correctly is very difficult. A team in the NLP group annotated about 250k words of web data with syntactic dependencies, and now we're looking for ways to automatically improve the quality of these annotations, mainly by identifying possible errors.

Automatically improving dependency annotation

Natalia Silveira natalias@stanford.edu

The project would to implement a classifier that detects annotation errors using features of the annotation, based on existing literature. This will allow students to learn a lot about dependency syntax, and understand some of the challenges involved in creating useful NLP resources -- as well as contribute to the NLP community, since this will be a widely distributed dataset.

Sublexical Compositionality

Jonathan Berant joberant@stanford.edu

Show slides

Nonsense Detector

Contact: Gabor angeli@stanford.edu

- Classify whether a sentence is a "well-formed sentence." In particular, the internet is full of a bunch of nonsense that we regularly treat as not-nonsense. This includes sentence fragments, headlines, YouTube comments, etc. The end result is that including these sentences tends to introduce noise in downstream tasks.

Nonsense Detector

Contact: Gabor angeli@stanford.edu

- The project would be to create a classifier to try to filter out these "bad" sentences -- for example, by looking for uncommon POS tag sequences, lack of punctuation (too much punctuation?), etc.

ML based grammar checker

Sida Wang

A good amount of data is collected on this from the CoNLL 2013 shared task

http://www.comp.nus.edu.sg/~nlp/conll13st.html

Examples that MS Word doesn't correct

Current approach with huge amount of specific features

ML based grammar checker

Observed article†	*the*
First word in NP†	*black*
Word i before ($i = 1, 2, 3$)†	{*on, sat, ..*}
Word i before NP ($i = 1, 2$)	{*on, sat, ..*}
Word + POS i before ($i = 1, 2, 3$)†	{*on+IN, sat+VBD, ..*}
Word i after ($i = 1, 2, 3$)†	{*black, door, ..*}
Word after NP	*period*
Word + POS i after ($N = 1, 2$)†	{*period+period, ..* }
Bag of words in NP†	{*black, door, mat*}
N-grams ($N = 2, .., 5$)‡	{*on_X, X_black, ..* }
Word before + NP†	*on+black_door_mat*
NP + N-gram after NP ($N = 1, 2, 3$)†	{ *black_door_mat+period, ..*}
Noun compound (NC)†	*door_mat*
Adj + NC†	*black+door_mat*

Semantic Parsing

Who did Humphrey Bogart marry in 1928?

Semantic Parsing

Who did Humphrey Bogart marry in 1928?

semantic parsing

$\text{Type.Person} \sqcap \text{Marriage.}(\text{Spouse.HumphreyBogart} \sqcap \text{StartDate.1928})$

Semantic Parsing

Who did Humphrey Bogart marry in 1928?

semantic parsing

Type.Person ⊓ Marriage.(Spouse.HumphreyBogart ⊓ StartDate.1928)

execute logical form

MaryPhilips

Semantic Parsing

Who did Humphrey Bogart marry in 1928?

semantic parsing

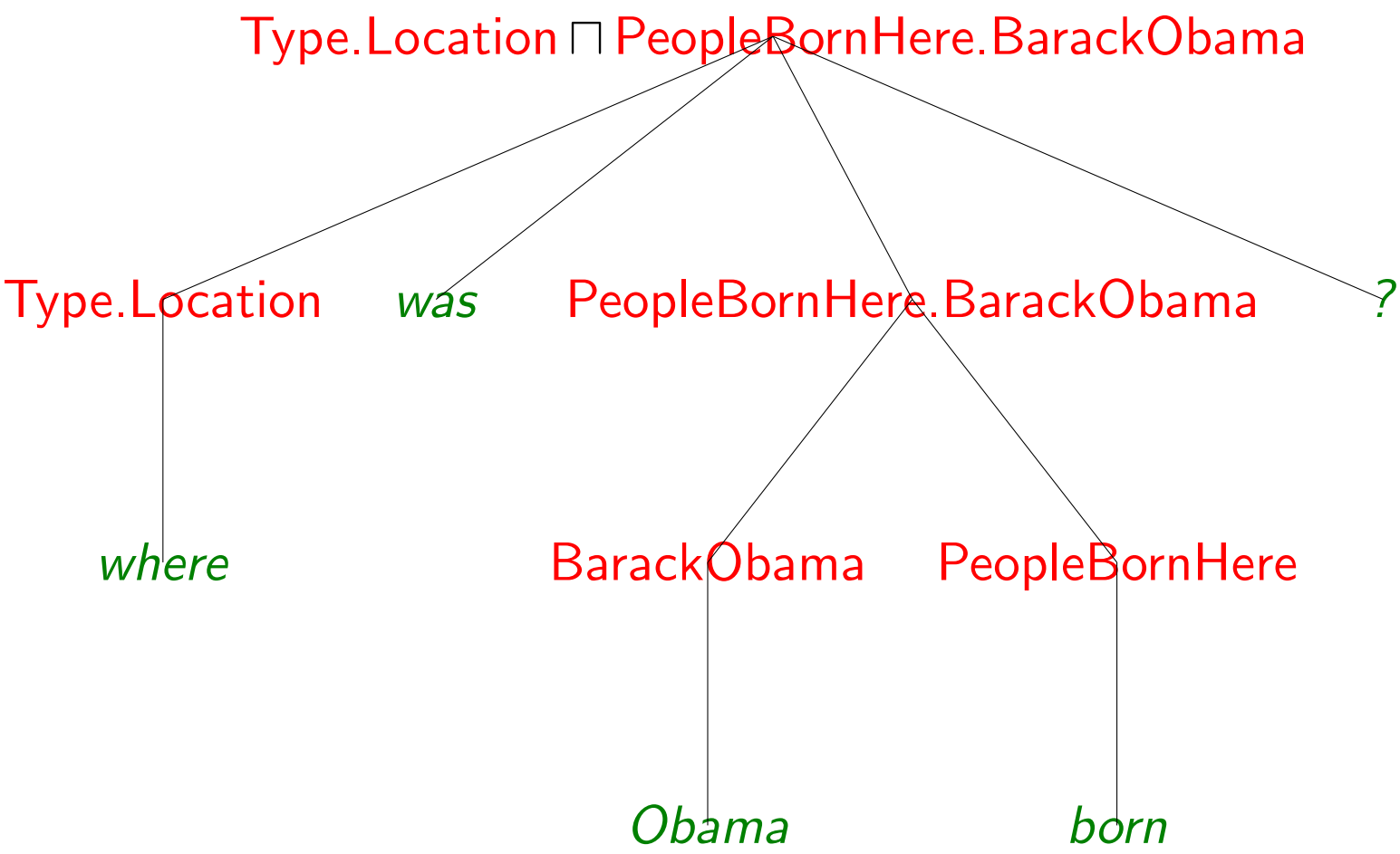
Type.Person ⊓ Marriage.(Spouse.HumphreyBogart ⊓ StartDate.1928)

execute logical form

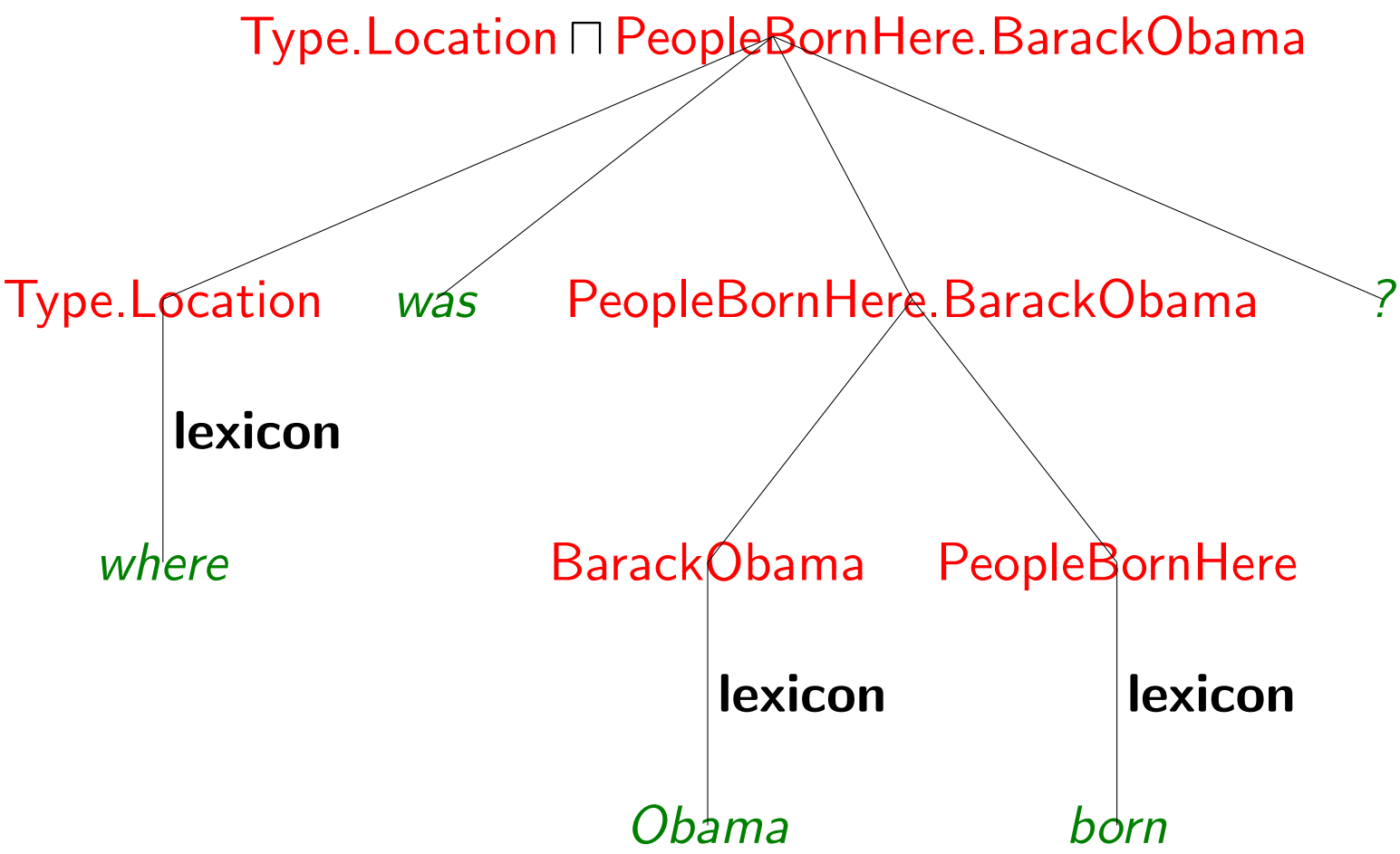
MaryPhilips

Motivation:: Natural language interface to large structured knowledge-bases such as Google's knowledge graph.

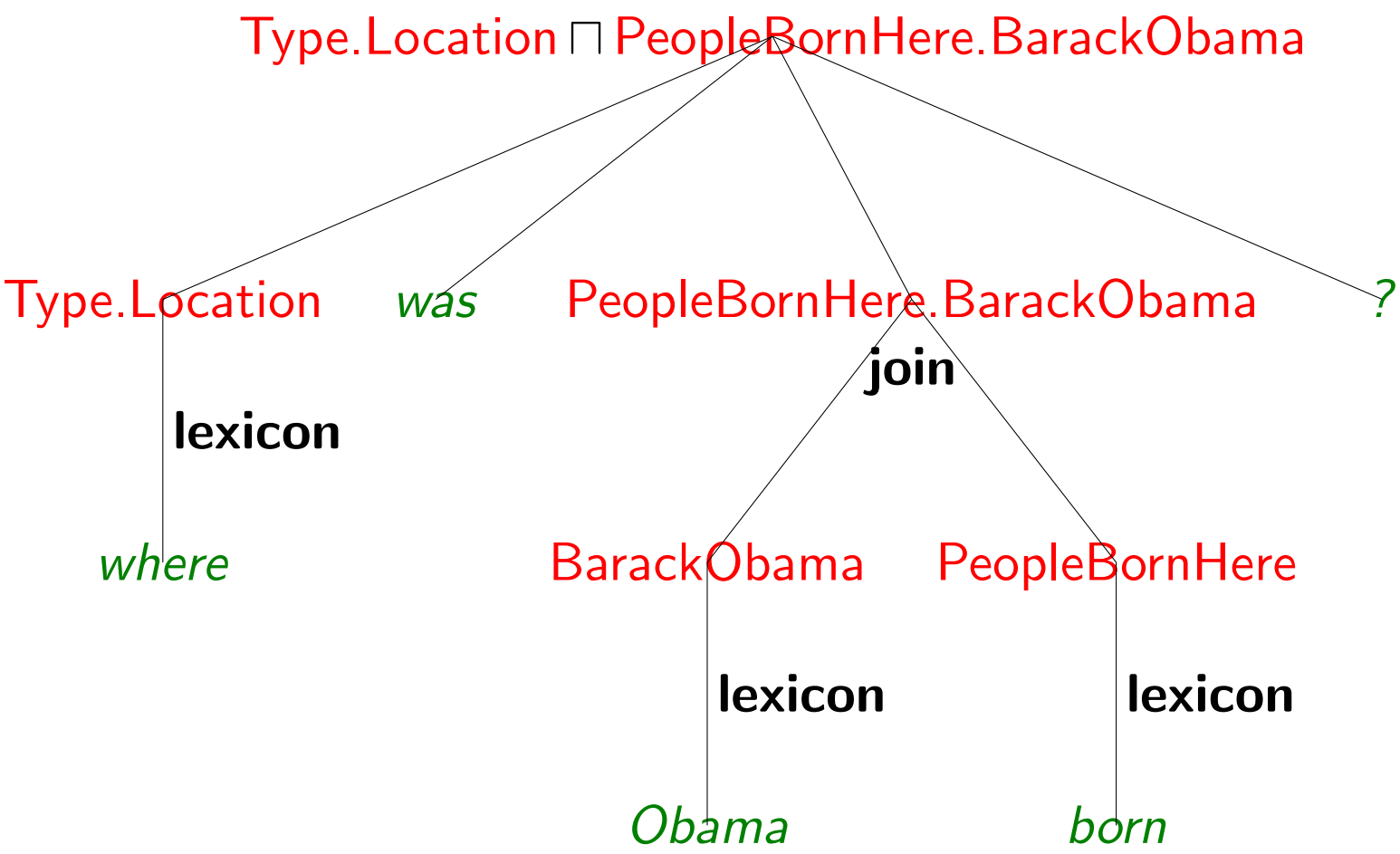
How is it done



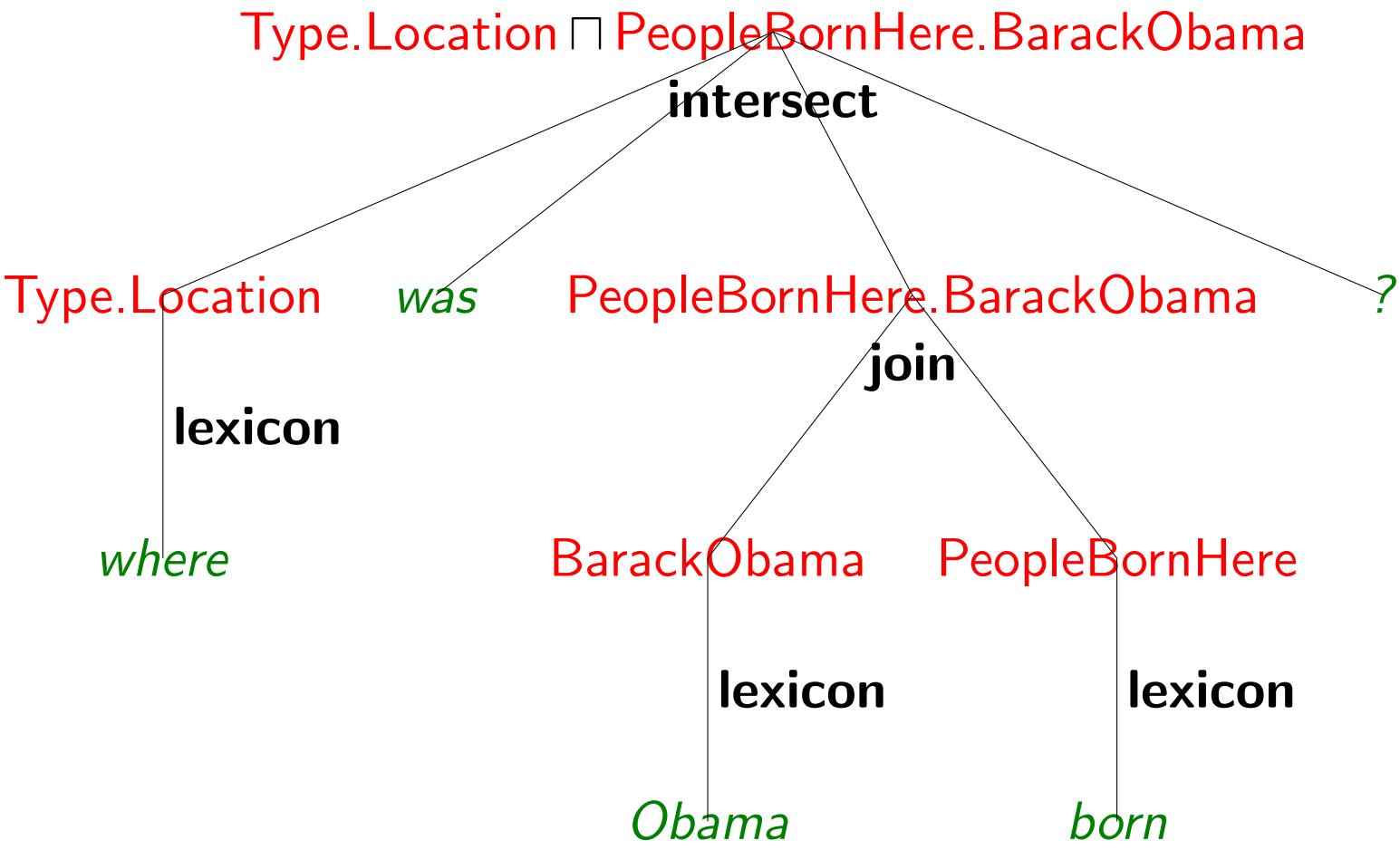
How is it done



How is it done



How is it done



Problem

Abstraction level in langauge and KB are different:

Result: lexicon should map text phrases to complex logical forms

Problem

Abstraction level in langauge and KB are different:

Result: lexicon should map text phrases to complex logical forms

actress$\Rightarrow$Gender.Female $\sqcap$ Profession.Actor

Problem

Abstraction level in langauge and KB are different:

Result: lexicon should map text phrases to complex logical forms

actress$\Rightarrow$Gender.Female $\sqcap$ Profession.Actor

grandfather$\Rightarrow$Gender.male $\sqcap$ Child.Child.X

Problem

Abstraction level in langauge and KB are different:

Result: lexicon should map text phrases to complex logical forms

actress$\Rightarrow$Gender.Female $\sqcap$ Profession.Actor

grandfather$\Rightarrow$Gender.male $\sqcap$ Child.Child.X

relative$\Rightarrow$Child.X $\sqcup$ Parent.X $\sqcup$ Sibling.X $\sqcup$...

Problem

Abstraction level in langauge and KB are different:

Result: lexicon should map text phrases to complex logical forms

actress$\Rightarrow$Gender.Female $\sqcap$ Profession.Actor

grandfather$\Rightarrow$Gender.male $\sqcap$ Child.Child.X

relative$\Rightarrow$Child.X $\sqcup$ Parent.X $\sqcup$ Sibling.X $\sqcup$...

mayor$\Rightarrow$Politician.(Position.Mayor $\sqcup$ Jurisdiction.X)

Project task

(*Barack Obama*, *was born in*, *Honolulu*)

(*Albert Einstein*, *was born in*, *Ulm*)

(*Barack Obama*, *lived in*, *Chicago*)

... 15M triples ...

Project task

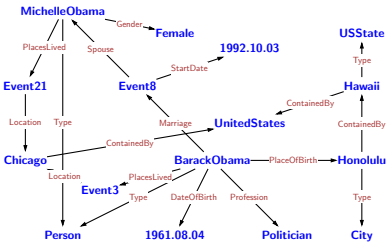
(*Barack Obama*, *was born in*, *Honolulu*)

(*Albert Einstein*, *was born in*, *Ulm*)

(*Barack Obama*, *lived in*, *Chicago*)

... 15M triples ...

Freebase:



(BarackObama, PlaceOfBirth, Honolulu)

(Albert Einstein, PlaceOfBirth, Ulm)

(BarackObama, PlacesLived.Location, Chicago)

... 600M triples ...

Project task

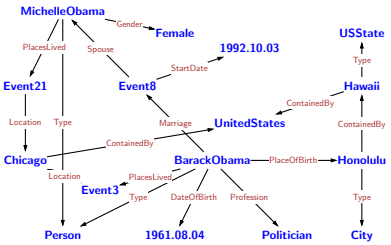
(*Barack Obama*, *was born in*, *Honolulu*)

(*Albert Einstein*, *was born in*, *Ulm*)

(*Barack Obama*, *lived in*, *Chicago*)

... 15M triples ...

Freebase:



(BarackObama, PlaceOfBirth, Honolulu)

(Albert Einstein, PlaceOfBirth, Ulm)

(BarackObama, PlacesLived.Location, Chicago)

... 600M triples ...

Align text phrases to complex KB predicates