Was Obama born in Kenya?

Obama's birthplace is not Kenya

No.
Distributional Semantics

- Induce the meanings of words from corpora
Distributional Semantics

- Induce the meanings of words from corpora

<table>
<thead>
<tr>
<th>X was born in Y</th>
<th>X's birthplace is Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Obama, Hawaii)</td>
<td>(Obama, Hawaii)</td>
</tr>
<tr>
<td>(Shakespeare, Stratford)</td>
<td>(Shakespeare, Stratford)</td>
</tr>
<tr>
<td>(Obama, 1961)</td>
<td>(Jesus, Bethlehem)</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

(e.g. Lin & Pantel, 2001)
\( \text{sim}(X \text{ was born in } Y, X's \text{ birthplace is } Y) > t \)

Obama was born in Hawaii

\( \Rightarrow \) Obama's birthplace is Hawaii
Distributional Semantics

Was Obama born in Kenya?

Obama's birthplace is not Kenya

Yes!
Formal Semantics

Obama
NP
was
(S\NP)/(S\NP)
born in
(S\NP)/NP
Hawaii
NP

S\NP

S\NP

S
Formal Semantics

<table>
<thead>
<tr>
<th>Obama</th>
<th>was</th>
<th>born in</th>
<th>Hawaii</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>(S\NP)/(S\NP)</td>
<td>(S\NP)/NP</td>
<td>NP</td>
</tr>
<tr>
<td>obama</td>
<td>$\lambda p \lambda x . p(x)$</td>
<td>$\lambda y \lambda x . \text{born_in}'(x, y)$</td>
<td>hawaii</td>
</tr>
</tbody>
</table>

\[
\text{S\NP} 
\]

\[
\text{S\NP} 
\]

\[
\text{S} 
\]
Formal Semantics

Obama
NP
obama

was
(S\NP)/(S\NP)
\( \lambda p \lambda x . p(x) \)

born in
(S\NP)/NP
\( \lambda y \lambda x . \text{born_in}'(x, y) \)

Hawaii
NP
hawaii

\[
\lambda x \ . \ \text{born_in}'(x, \text{hawaii})
\]

\[
\lambda x \ . \ \text{born_in}'(x, \text{hawaii})
\]

S
\[
\text{born_in}'(\text{obama}, \text{hawaii})
\]
Formal Semantics

Obama

NP

obama


wasn't

(S\NP)/(S\NP)

λpλx . ¬p(x)


born in

(S\NP)/NP

λyλx . born_in'(x, y)


Kenya

NP

kenya


canada

(S\NP)

λx . ¬born_in'(x, canada)


(S\NP)

λx . ¬born_in'(x, hada)


S

S

¬born_in'(obama, canada)

¬born_in'(obama, hada)
Obama wasn't born in Kenya

⇒

Obama was born in Kenya
Every US president was born in the United States

⇒

Obama wasn't born in Kenya
Nicole Kidman and Barack Obama were born in Hawaii

⇔

Nicole Kidman was born in Hawaii and Barack Obama was born in Hawaii
Formal Semantics

Was Obama born in Kenya?

Obama's birthplace is not Kenya

Don't know.
¬birthplace'(obama, kenya)
⇒?
born_in'(obama, kenya)
Formal Semantics

+ Models semantic operators ("not", "every", etc.)

+ Abstracts away from syntax (conjunctions, passive voice, relative clauses, etc)

- Doesn't model meaning of content words
Combining Formal and Distributional Semantics

Existing approaches:

● Compositional vector space models
  ○ (e.g. Coecke et al., 2011, Baroni et al., 2012, Grefenstette 2013)

● Learn inference rules as logical axioms
  ○ (e.g. Garrette et al., 2011, Beltagy et al., 2013)
Obama was born in Hawaii

born_in'(obama, hawaii)

Hawaii is Obama's birthplace

birth_place'(obama, hawaii)
Combining Formal and Distributional Semantics

*Obama was born in Hawaii*

relation53(Obama, Hawaii)

*Hawaii is Obama's birthplace*

relation53(Hawaii, Obama's birthplace)
Learn lexical entries such that:

\[
\text{born} \vdash (S\backslash NP)/PP[\text{in}] : \lambda x\lambda y \cdot \text{relation53}(x,y)
\]

\[
\text{birthplace} \vdash N/PP[\text{of}] : \lambda x\lambda y \cdot \text{relation53}(x,y)
\]
Formal Semantics

Obama 
NP 

obama 

was 
(S\NP)/(S\NP) 

\( \lambda p \lambda x . p(x) \)

born in 
(S\NP)/NP 

\( \lambda y \lambda x . \text{relation53}(x, y) \)

Hawaii 
NP 

hawaii 

\[ \lambda x . \text{relation53}(x, \text{hawaii}) \]

\[ \lambda x . \text{relation53}(x, \text{hawaii}) \]

\[ \text{relation53}(\text{obama}, \text{hawaii}) \]
Combining Formal and Distributional Semantics

Gather statistics on predicates in large corpus:

<table>
<thead>
<tr>
<th>born_in'(X, Y)</th>
<th>birthplace_of'(X, Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Obama, Hawaii)</td>
<td>(Obama, Hawaii)</td>
</tr>
<tr>
<td>(Shakespeare, Stratford)</td>
<td>(Shakespeare, Stratford)</td>
</tr>
<tr>
<td>(Obama, 1961)</td>
<td>(Jesus, Bethlehem)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Combining Formal and Distributional Semantics

born_in'(X, Y)  employeeOf'(X, Y)
birth_place(X,Y)  workFor'(X, Y)

buy'(X, Y)  write'(X, Y)
purchase'(X, Y)  authorOf'(X,Y)
acquisitionOf'(X, Y)
Combining Formal and Distributional Semantics

- born_in'(X, Y)
- birth_place(X, Y)
- buy'(X, Y)
- purchase'(X, Y)
- acquisitionOf'(X, Y)
- employeeOf'(X, Y)
- workFor'(X, Y)
- write'(X, Y)
- authorOf'(X, Y)
Combining Formal and Distributional Semantics

Cluster using Chinese Whispers (Biemann, 2006)
Was Obama born in Kenya?

Obama's birthplace is not Kenya

No.
Ambiguity

Obama was born in Hawaii
Obama was born in 1961
Obama was born in 1961

\[\nequiv\]

Obama's birthplace is 1961
Create multiple typed lexical entries:

\[ \text{born} \vdash (S\backslash NP) / PP[\text{in}] : \lambda y \, \lambda x \, . \, \text{birthPlace}(x, y) \]
\[ \text{born} \vdash (S\backslash NP) / PP[\text{in}] : \lambda y \, \lambda x \, . \, \text{birthDate}(x, y) \]

(similar to: Schoenmackers et al., 2010; Berant et al., 2011)
Clustering Typed Predicates

Only cluster predicates with the same type

born_in(X:PER, Y:DAT)
birth_date(X:PER, Y:DAT) \rightarrow \text{relation53}(X, Y)

born_in(X:PER, Y:LOC)
birth_place(X:PER, Y:LOC)
native_of(X:PER, Y:LOC) \rightarrow \text{relation91}(X, Y)
Topic Model

One 'document' per predicate
One 'word' per argument

<table>
<thead>
<tr>
<th>lives in X</th>
<th>year of X</th>
<th>born in X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hawaii</td>
<td>1564</td>
<td>Hawaii</td>
</tr>
<tr>
<td>London</td>
<td>2001</td>
<td>2001</td>
</tr>
<tr>
<td>France</td>
<td>1945</td>
<td>London</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1564</td>
</tr>
</tbody>
</table>
Topic Model

One 'document' per predicate
One 'word' per argument

<table>
<thead>
<tr>
<th>lives in X</th>
<th>year of X</th>
<th>born in X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hawaii</td>
<td>1564</td>
<td>Hawaii</td>
</tr>
<tr>
<td>London</td>
<td>2001</td>
<td>2001</td>
</tr>
<tr>
<td>France</td>
<td>1945</td>
<td>London</td>
</tr>
</tbody>
</table>

(see also: Melamud et al., 2013)
### Example Types

<table>
<thead>
<tr>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>suspect, assailant, fugitive, accomplice, ...</td>
</tr>
<tr>
<td>author, singer, actress, actor, dad, ...</td>
</tr>
<tr>
<td>city, area, country, region, town, capital, ...</td>
</tr>
<tr>
<td>subsidiary, automaker, airline, Co., GM, ...</td>
</tr>
<tr>
<td>musical, thriller, sequel, special, ...</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Combining types

\[\lambda y:\left\{\begin{array}{l}
LOC \sim 0.5 \\
DATE \sim 0.4 \\
...\end{array}\right\}, \lambda x:\left\{\begin{array}{l}
PER \sim 0.9 \\
IDEA \sim 0.05 \\
...\end{array}\right\}, \text{born}_\text{in}(x, y)\]  

\[2001:\left\{\begin{array}{l}
\text{DATE} \sim 0.8 \\
\text{FILM} \sim 0.15 \\
...\end{array}\right\}\]
Combining types

\[
\frac{(S\backslash NP)}{NP} \quad \lambda y : \quad \lambda x : \quad . \text{born_in}(x, y) \quad \frac{2001}{NP}
\]

\[
\begin{align*}
\lambda y : & \left\{ \begin{array}{l}
\text{LOC} \sim 0.5 \\
\text{DATE} \sim 0.4 \\
\text{...}
\end{array} \right. \\
\lambda x : & \left\{ \begin{array}{l}
\text{PER} \sim 0.9 \\
\text{IDEA} \sim 0.05 \\
\text{...}
\end{array} \right. \\
& \text{born_in}(x, y) \\
& 2001 : \left\{ \begin{array}{l}
\text{DATE} \sim 0.8 \\
\text{FILM} \sim 0.15 \\
\text{...}
\end{array} \right.
\end{align*}
\]

\[
(S\backslash NP) \quad \lambda x : \left\{ \begin{array}{l}
\text{PER} \sim 0.9 \\
\text{IDEA} \sim 0.05 \\
\text{...}
\end{array} \right. \\
. \text{born_in}(x, 2001 : \left\{ \begin{array}{l}
\text{DATE} \sim 0.95 \\
\text{LOC} \sim 0.02 \\
\text{...}
\end{array} \right.)
\]
Combining types

\[ \lambda y: \begin{cases} 
  \text{LOC} \sim 0.5 \\
  \text{DATE} \sim 0.4 \\
  \text{...} 
\end{cases} \quad \lambda x: \begin{cases} 
  \text{PER} \sim 0.9 \\
  \text{IDEA} \sim 0.05 \\
  \text{...} 
\end{cases} \quad \text{born}_\text{in}(x, y) \]

Washington
\[ \begin{cases} 
  \text{LOC} \sim 0.8 \\
  \text{PER} \sim 0.15 \\
  \text{...} 
\end{cases} \]
Combining types

\[ \lambda y: \{ \text{LOC} \sim 0.5, \text{DATE} \sim 0.4, \ldots \} \quad \lambda x: \{ \text{PER} \sim 0.9, \text{IDEA} \sim 0.05, \ldots \} \cdot \text{born\_in}(x, y) \]

\[ \text{Washington} \]

\[ \text{NP} \quad \text{NP} \]

\[ \lambda x: \{ \text{PER} \sim 0.9, \text{IDEA} \sim 0.05, \ldots \} \cdot \text{born\_in}(x, \text{Wash}: \{ \text{LOC} \sim 0.8, \text{PER} \sim 0.15, \ldots \}) \]

\[ \text{S\textbackslash NP} \]

\[ \lambda x: \{ \text{PER} \sim 0.9, \text{IDEA} \sim 0.05, \ldots \} \cdot \text{born\_in}(x, \text{Wash}: \{ \text{LOC} \sim 0.95, \text{PER} \sim 0.01, \ldots \}) \]
Compositional Semantics

\[ \text{born} \vdash (S \backslash NP)/PP[\text{in}] : \lambda y_{\text{LOC}} \lambda x_{\text{PER}} . \text{birthPlace}(x, y) \]

\[ \text{born} \vdash (S \backslash NP)/PP[\text{in}] : \lambda y_{\text{DAT}} \lambda x_{\text{PER}} . \text{birthDate}(x, y) \]
born ⊬ (S\NP)/PP[\text{in}] : \lambda y_{\text{LOC}} \lambda x_{\text{PER}} . \text{birthPlace}(x, y)

born ⊬ (S\NP)/PP[\text{in}] : \lambda y_{\text{DAT}} \lambda x_{\text{PER}} . \text{birthDate}(x, y)

\begin{align*}
\text{born} \vdash (S\NP)/PP[\text{in}] \colon \lambda y \lambda x . & \begin{cases}
(x:\text{LOC}, y:\text{PER}) \Rightarrow \text{birthPlace}(x, y) \\
(x:\text{DAT}, y:\text{PER}) \Rightarrow \text{birthDate}(x, y)
\end{cases}
\end{align*}
Compositional Semantics

Output packed logical form capturing joint distribution:

\[
\begin{align*}
\text{Obama was born in Hawaii in 1961} & \quad \{ \text{birthPlace} \sim 0.9, \text{birthDate} \sim 0.1 \} \quad \text{(Obama, Hawaii)} \\
\land & \quad \{ \text{birthPlace} \sim 0.1, \text{birthDate} \sim 0.9 \} \quad \text{(Obama, 1961)}
\end{align*}
\]
Training Details

- Train on English Gigaword
- Use C&C parser for syntax (Clark and Curran 2004)
- 15 entity types
- Lexicon includes manual entries for some function words ("every", "not", etc.)
Evaluation

Automatically generate question set from corpus

Google bought YouTube

Who bought YouTube?
What did Google buy?
Evaluation

Automatically generate question set from corpus

Google bought YouTube

Who bought YouTube?
What did Google buy?

Generate 1000 questions
Manual evaluation
Comparison Systems

- Reverb (Fader et al., 2011)
- Relational LDA (Yao et al., 2011)
- CCG Baseline
- CCG + WordNet
- CCG using Distributional Clusters
Examples

**Question:** What did Delta merge with?

**Sentence:** The 747 freighters came with Delta's acquisition of Northwest

**Answer:** Northwest
Evaluating Formal Semantics

Problems involving inference using quantifiers from the FraCaS Suite (Section 1):

**Premises:**
Every European has the right to live in Europe.
Every European is a person.
Every person who has the right to live in Europe can travel freely within Europe.

**Hypothesis:**
Every European can travel freely within Europe.
## Evaluating Formal Semantics

<table>
<thead>
<tr>
<th></th>
<th>Single Premise Sentence</th>
<th>Multiple Premise Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Logic 2007</td>
<td>84%</td>
<td>-</td>
</tr>
<tr>
<td>Natural Logic 2008</td>
<td>98%</td>
<td>-</td>
</tr>
<tr>
<td>CCG-Dist (parser)</td>
<td>70%</td>
<td>50%</td>
</tr>
<tr>
<td>CCG-Dist (gold syntax)</td>
<td>89%</td>
<td>80%</td>
</tr>
</tbody>
</table>
Conclusions

- Formal and distributional semantics complement each other
- Modelling content words using cluster identifiers gives the benefits of both
Thanks!

Any questions?