Combined Distributional and Logical Semantics

Yiyi Chen

CIS, LMU Munich

18th July 2019

Motivation

- The new postdoc doesn't work for Kim: she writes papers on semantics. And.. uh.. on these neural nets that have loops.

Motivation

- The new postdoc doesn't work for Kim: she writes papers on semantics. And.. uh.. on these neural nets that have loops.
  - descriptive content of words, e.g.*postdoc* or *write* (i.e. have knowledge of the properties of the involved individuals and events)

Motivation

- The new postdoc doesn't work for Kim: she writes papers on semantics. And.. uh.. on these neural nets that have loops.
  - descriptive content of words, e.g.*postdoc* or *write* (i.e. have knowledge of the properties of the involved individuals and events)
  - function words and structural markers, e.g. *the*, *and*

Motivation

- The new postdoc doesn't work for Kim: she writes papers on semantics. And.. uh.. on these neural nets that have loops.
    - descriptive content of words, e.g.*postdoc* or *write* (i.e. have knowledge of the properties of the involved individuals and events)
    - function words and structural markers, e.g. *the*, *and*
    - singular/plural constructions a) a unique postdoc, b) a person named Kim and c) a number of entities that are papers, and d) abstract concepts refered to generically (*semantics*, *neural nets*)

Motivation

- The new postdoc doesn't work for Kim: she writes papers on semantics. And.. uh.. on these neural nets that have loops.
  - descriptive content of words, e.g.*postdoc* or *write* (i.e. have knowledge of the properties of the involved individuals and events)
  - function words and structural markers, e.g. *the*, *and*
  - singular/plural constructions a) a unique postdoc, b) a person named Kim and c) a number of entities that are papers, and d) abstract concepts refered to generically (*semantics*, *neural nets*)
  - composition over the constituents of each sentence, shifting the meanings of *write* and *paper* towards *authoring* and *scientific article*

Motivation

- The new postdoc doesn't work for Kim: she writes papers on semantics. And.. uh.. on these neural nets that have loops.
  - descriptive content of words, e.g.*postdoc* or *write* (i.e. have knowledge of the properties of the involved individuals and events)
  - function words and structural markers, e.g. *the*, *and*
  - singular/plural constructions a) a unique postdoc, b) a person named Kim and c) a number of entities that are papers, and d) abstract concepts refered to generically (*semantics*, *neural nets*)
  - composition over the constituents of each sentence, shifting the meanings of *write* and *paper* towards *authoring* and *scientific article*
  - inferences, for instance: the new postdoc is knowledgeable about semantics and what is Recurrent Neural Networks; she cannot be working for Kim, who is a syntactician; but she perhaps works for Sandy, who does Machine Learning.

Motivation

- No single semantic framework has been proposed that would naturally cater to all these phenomena.

Formal Semantics

- Formal semantics adopts some of the tools standardly used to describe formal languages
  - it caters to ontological matters (what there is in the world)
  - reference to entities (how we talk *about* things)
  - meaning at the higher constituent level (composition)
  - interpretation at the sentential level (by giving propositions a truth value)
  - logical inferences that can be drawn from a particular sentence

Formal Semantics - describe the world via a model

- Model-theoretic, truth-conditional semantics interprets linguistic expressions with respect to some model.
- A model is represented in terms of sets.
- For instance, a world with three postdocs will contain a set of three sharing the property of being a *postdoc*.

Formal Semantics - reference to entities

- *extension* and *intension*
- *extension* - the entities it refers to in the model
  - the extension of *postdoc* is the set of all postdocs in the model
- *intension* - a function mapping possible worlds to extensions.

Formal Semantics - model **compositionality**

- The meaning of a complex expression is derivable from the meaning of its parts in a systematic and productive way.

Formal Semantics - model **compositionality**

- The meaning of a complex expression is derivable from the meaning of its parts in a systematic and productive way.
- A compositional formal semantics framework provides semantic representations of linguistic expressions in a logic, and rules for combining them.

Formal Semantics - model **compositionality**

- The meaning of a complex expression is derivable from the meaning of its parts in a systematic and productive way.
- A compositional formal semantics framework provides semantic representations of linguistic expressions in a logic, and rules for combining them.
- For any expression, it is possible to identify a number of potentially distinct individuals and the relationships between them.

Formal Semantics - model **compositionality**

- The meaning of a complex expression is derivable from the meaning of its parts in a systematic and productive way.
- A compositional formal semantics framework provides semantic representations of linguistic expressions in a logic, and rules for combining them.
- For any expression, it is possible to identify a number of potentially distinct individuals and the relationships between them.
- the sentence *The new postdoc has written several articles* can be transformed into the following logical form:
  - $\exists x, y[\, new(postdoc(\, x)\,) \wedge article * (\, y) \wedge write(\, x, y)\,]$

Formal Semantics - model **compositionality**

- The meaning of a complex expression is derivable from the meaning of its parts in a systematic and productive way.
- A compositional formal semantics framework provides semantic representations of linguistic expressions in a logic, and rules for combining them.
- For any expression, it is possible to identify a number of potentially distinct individuals and the relationships between them.
- the sentence *The new postdoc has written several articles* can be transformed into the following logical form:
  - $\exists x, y[\, new(postdoc(\, x)\,) \wedge article * (\, y) \wedge write(\, x, y)\,]$
- A complete account of compositionality relies on being able to interpret function words in the sentence.

Formal Semantics - model **compositionality**

- The meaning of a complex expression is derivable from the meaning of its parts in a systematic and productive way.
- A compositional formal semantics framework provides semantic representations of linguistic expressions in a logic, and rules for combining them.
- For any expression, it is possible to identify a number of potentially distinct individuals and the relationships between them.
- the sentence *The new postdoc has written several articles* can be transformed into the following logical form:
  - $\exists x, y[\, new(postdoc(\, x)\,) \wedge article * (\, y) \wedge write(\, x, y)\,]$
- A complete account of compositionality relies on being able to interpret function words in the sentence.
- Formalization of quantifiers (also negations, modals), to describe entities with assigning them properties.

Formal Semantics - the meaning of a sentence

- In truth-theoretic semantics, the meaning of a sentence is a function from possible worlds to **truth values**.
- 'Snow is white' is true if and only if snow is white. (Tarski 1994)
- the notion of *satisfaction*
- A predicate can be truthfully applied to a term if the corresponding property in a world applies to the referent of the term.

Formal Semantics - inference

- Being able to give an interpretation to a sentence with respect to truth or speaker belief is an essential part of explaining the implicit aspects of meaning, in particular **inference**.
- infer from *a postdoc is writing* to *a human is writing*
- proof theory provides classes of inference that are algorithmically decidable.
- a postdoc may be cast as an individual entity which is a human, holds a doctorate

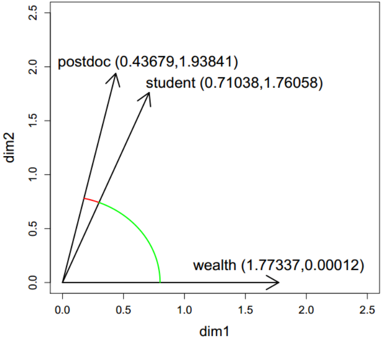
Formal Semantics - problems

- Fail to represent content words in their richness.
  - the distinction between near-synonyms such as *man/gentlemen/chap/lad/guy/dude/bloke*
- Formal semantics limits what it encodes of human linguistic experience.
  - analogical reasoning
- Human knowledge is not always certain and consistent, nevertheless, people can communicate with each other.

Distributional Semantics

- the meaning representation for a linguistic expression is a function of the contexts in which it occurs.
- **contexts**: the words surrounding the target word, syntactic relations, etc.
- Distributional representations are vectors or more complex algebraic objects such as matrices and tensors
- The semantic information is distributed across all the dimensions of the vector
- It is encoded in the form of **continuous** values, which allows for very rich and nuanced information.

Distributional Semantics



- the collection of words in a lexicon forms a vector space or semantic space, in which semantic relations can be modeled as geometric relations.

Distributional Semantics - human judgements

- explore synonymy: puma-cougar
- noun categorization: car IS-A vehicle; banana IS-A fruit
- selectional preference: eat topinambur vs. *eat sympathy
- analogy: masion is to stone like carpenter is to wood
- relation classification: exam-anxiety: CAUSE-EFFECT

Distributional Semantics - meaning nuances

Table 1
Near-synonyms in semantic space: The words closest to *man*, *chap*, *lad*, *dude*, and *guy* in the distributional model of Baroni, Dinu, and Kruszewski (2014), based on the UKWaC. Taken from Baroni (2016).

man	**chap**	**lad**	**dude**	**guy**
woman	bloke	boy	freakin’	bloke
gentleman	guy	bloke	woah	chap
gray-haired	lad	scouser	dorky	doofus
boy	fella	lass	dumbass	dude
person	man	youngster	stoopid	fella

Distributional Semantics - compositionality

- Compositional Distributional Semantics allows us to model semantic phenomena that are very challenging for Formal Semantics.
- polysemy
  - During that **postdoc**, I didn't publish anything. (event)
  - The **postdoc** shook her head. (human)

Formal Distributional Semantics - Case Study

Input Sentence

Shakespeare wrote Macbeth

$\Downarrow$

Intial semantic analysis

$write_{arg0,arg1}(shakespeare,\ macbeth)$

$\Downarrow$

Entity Typing

$write_{arg0:PER,arg1:BOOK}(shakespeare{:}PER,\ macbeth{:}BOOK)$

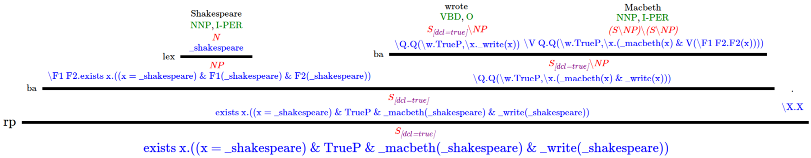
$\Downarrow$

Distributional semantic analysis

$relation37(shakespeare{:}PER,\ macbeth{:}BOOK)$

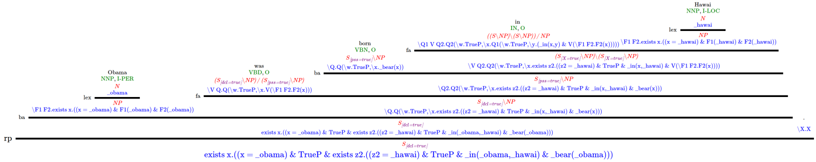
Figure 2: Layers used in our model.

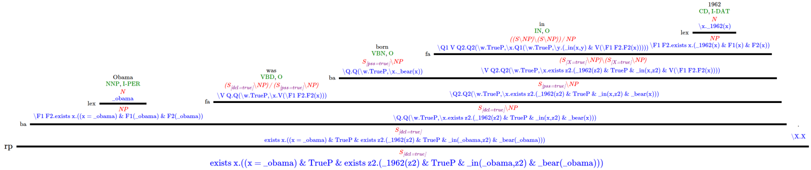
Case Study - Initial Semantic Analysis

- 

- mapping sentences into logical forms: $\exists x \exists y(\mathit{Shakespeare}(x) \wedge \mathit{Macbeth}(y) \wedge \mathit{write}(x, y))$
- extract the predicate-argument structure from the logical form: $\mathit{write}_{arg0,arg1}(\mathit{shakespeare}, \mathit{macbeth})$

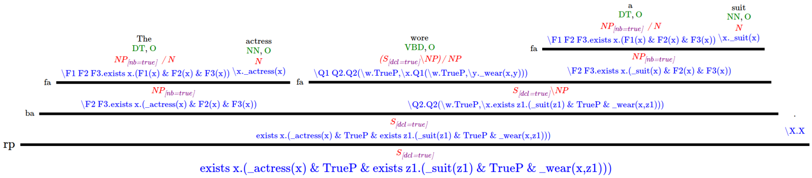
Case Study - Disambiguating Polysemous Predicates

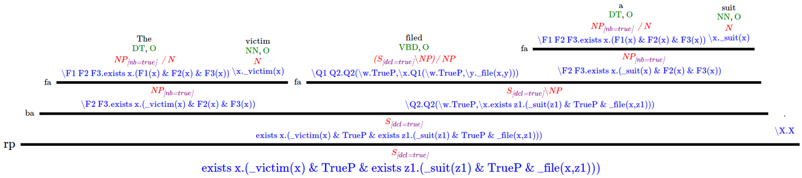
- 

- 

- to disambiguate, we use typed predicates:
- $born_{arg0:PER,argIn:DAT}$ and $born_{arg0:PER,argIn:LOC}$

Case Study - Disambiguation

- 

- $\exists x \exists y (\, actress(\, x \,) \wedge suit(\, y \,) \wedge wear(\, x, y \,) \,)$
- 

- $\exists x \exists y (\, victim(\, x \,) \wedge suit(\, y \,) \wedge file(\, x, y \,) \,)$
- to disambiguate, we use typed predicates:
- $wear_{arg0:PER,arg1:CLOTHES}$ and $file_{arg0:PER,arg1:LEGAL}$

Case study - Distributional Relation Clustering

-

	(Shakespeare, Macbeth)	(Dickens, Oliver Twist)	(Rowling, Harry Potter)	(Council, Dublin)	(Sharon, Ronnie)
write	10	20	10	0	0
suggest	0	0	0	10	5

- The type of a predicate must match the type of its arguments, so the type distribution of a binary predicate is simply the joint distribution of the two argument type distributions.
- $write_{arg0,arg1}(Shakespeare, Macbeth)$,with *shakespeare* having type distribution (PER=0.9, LOC=0.1) and *Macbeth* (BOOK:0.7, DAT=0.3), etc.
-

	(Shakespeare, Macbeth)	(Dickens, Oliver Twist)	(Rowling, Harry Potter)	(Council, Dublin)	(Sharon, Ronnie)
write_{arg0:PER,arg1:Book}	10.63	20.63	10.63	0.00	0.27
suggest_{arg0:ORG, arg1:LOC}	0.25	0.25	0.25	10.12	0.63
suggest_{arg0:PER, arg1:PER}	0.25	0.25	0.25	0.12	5.63

Implementation

	(afrikaans, dutch)	(dione, dodona)	(council of whitby, english history)	(yasujiro ozu, late spring)	(russia, brahmos)	(fiji, kingdom of tonga)	(world cup, olympic football)	(dublin, county donegal)	(carpenter, escape from new york)	(norway, ethiopian)	...	(fryderyk chopin, warsaw lyceum)	(motorola, apple)	(vax, vms)	(iraq, persian gulf)
star	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
be	1	1	0	0	0	1	0	0	0	0	...	0	0	0	0
appoint	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
established	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
carry	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
give	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
hold	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
express	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
have	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
consist	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
play	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
remain	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0
lose	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0

- 1418 predicates, 14859 argument pairs

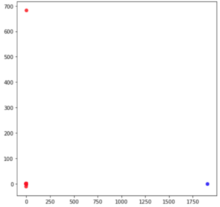
Implementation

	(boc tower, hong kong)	(chaplin, chaplin)	(walter gropius, bauhaus)	(hawaii state supreme court, ali ʻ iōlani hale)	(thurgot, saint andrews)	(santiniketan, indian)	(goldwater, reagan)	(dostum, rabbani)	(coloman, béla)	(rockies, oakland athletics)	...	(bali strait, java)	(brown, sarah macaulay)	(statenvertaling, dutch)	(sagan, aaas 's)	(spain, coimbra)	(france, cascade mountains)	(love, brian wilson)	(lenin, reds)	(kelly, daisy bell)	(mclean, lisa kindred)
star-PER-MISC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.00	0.00	0.00	0.0	1.0	0.0
star-PER-PER	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	0.00	0.38	0.0	0.0	0.00	0.00	0.78	0.0	0.0	1.0
star-ORG-PER	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.62	0.0	0.0	0.00	0.00	0.00	0.0	0.0	0.0
star-MISC-MISC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	1.0	0.0	0.00	0.00	0.00	0.0	0.0	0.0
star-MISC-PER	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.00	0.00	0.22	0.0	0.0	0.0
star-LOC-MISC	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.43	0.00	0.0	0.0	0.00	0.00	0.00	0.0	0.0	0.0
star-PER-LOC	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.00	0.04	0.00	0.1	0.0	0.0
star-PER-ORG	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	1.0	0.00	0.00	0.00	0.9	0.0	0.0
be-ORG-ORG	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.00	0.00	0.0	0.0	0.00	0.00	0.00	0.0	0.0	0.0
be-LOC-MISC	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.43	0.00	0.0	0.0	0.00	0.00	0.00	0.0	0.0	0.0
be-PER-MISC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.00	0.00	0.00	0.0	1.0	0.0
be-PER-LOC	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.00	0.04	0.00	0.1	0.0	0.0
be-ORG-LOC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.03	0.00	0.00	0.0	0.0	0.0
be-PER-PER	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	...	0.00	0.38	0.0	0.0	0.00	0.00	0.78	0.0	0.0	1.0

- 4560 typed predicates, 14859 argument pairs

Result - Clustering Untyped Predicates

```
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

X_pred = StandardScaler().fit_transform(features_pred)


pca4 = PCA(n_components=4)
x_4d = pca4.fit_transform(X_pred)

#Set a 3 KMeans clustering
kmeans_pred = KMeans(n_clusters = 3)

#Compute cluster centers and predict cluster indices
X_clustered = kmeans_pred.fit_predict(x_4d)

#Define our own color map
LABEL_COLOR_MAP = {0:'r', 1: 'b', 2: 'purple'}
label_color = [LABEL_COLOR_MAP[l] for l in X_clustered]

# Plot the scatter digram
plt.figure(figsize = (7,7))
plt.scatter(x_4d[:,0],x_4d[:,2], c= label_color, alpha=0.8)
plt.show()
```



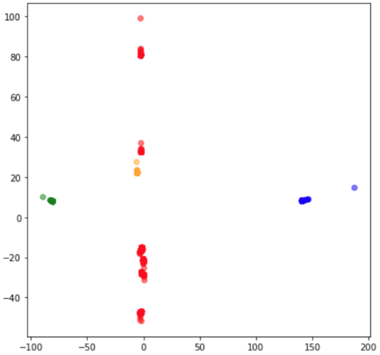
- cluster 1, 1345 predicates
-
	data_index	cluster
0	star	0
3	established	0
4	carry	0
5	give	0
7	express	0
10	play	0
13	beat	0
14	clash	0
15	establish	0
16	leave	0
17	snub	0
18	bury	0
19	incorporate	0
20	employ	0
22	write	0
23	serve	0
24	bear	0
25	introduce	0
26	condemn	0
27	oppose	0
28	repeal	0

- cluster 2, 1 predicate
-
	data_index	cluster
1	be	1

- cluster 3, 72 predicates
-
	data_index	cluster
2	appoint	2
6	hold	2
8	have	2
9	consist	2
11	remain	2
12	lose	2
21	lead	2
30	follow	2
34	manage	2
36	link	2
37	become	2
38	provide	2
41	form	2
55	begin	2

Result - Clustering typed Predicates

```
X = StandardScaler().fit_transform(features_df)

pca4 = PCA(n_components=16)
x_4d = pca.fit_transform(X)

#Set a 3 KMeans clustering
kmeans = KMeans(n_clusters = 4)

#Define our own color map
X_clustered = kmeans.fit_predict(x_4d)
LABEL_COLOR_MAP = {0:'r', 1: 'g', 2: 'b', 3:'orange'}
label_color = [LABEL_COLOR_MAP[l] for l in X_clustered]

# Plot the scatter digram
plt.figure(figsize = (7,7))
plt.scatter(x_4d[:,0],x_4d[:,2],c= label_color, alpha=0.5, cmap='viridis')
plt.show()
```



Result - Clustering typed Predicates

- cluster 1, 425 typed predicates
-

	data_index	cluster
16	be-LOC-LOC	0
35	established-LOC-LOC	0
48	carry-LOC-LOC	0
70	give-LOC-LOC	0
74	hold-LOC-LOC	0
93	have-LOC-LOC	0
106	consist-LOC-LOC	0
121	play-LOC-LOC	0
129	remain-LOC-LOC	0
145	lose-LOC-LOC	0
161	beat-LOC-LOC	0
165	clash-LOC-LOC	0
173	establish-LOC-LOC	0
185	leave-LOC-LOC	0
196	incorporate-LOC-LOC	0
213	lead-LOC-LOC	0
233	write-LOC-LOC	0
244	serve-LOC-LOC	0
300	follow-LOC-LOC	0
315	offer-LOC-LOC	0
322	separate-LOC-LOC	0
331	manage-LOC-LOC	0
335	cause-LOC-LOC	0
341	link-LOC-LOC	0
348	become-LOC-LOC	0
365	provide-LOC-LOC	0
377	continue-LOC-LOC	0
390	form-LOC-LOC	0
410	feature-LOC-LOC	0
417	contribute-LOC-LOC	0
...	...	...
4173	categorize-LOC-LOC	0
4174	rent-LOC-LOC	0
4177	rest-LOC-LOC	0
4196	double-LOC-LOC	0
4204	centre-LOC-LOC	0
4209	depart-LOC-LOC	0
4211	resolve-LOC-LOC	0

- cluster 2, 3053 typed predicates
-

	data_index	cluster
2	star-ORG-PER	1
3	star-MISC-MISC	1
4	star-MISC-PER	1
5	star-LOC-MISC	1
6	star-PER-LOC	1
7	star-PER-ORG	1
8	be-ORG-ORG	1
9	be-LOC-MISC	1
11	be-PER-LOC	1
12	be-ORG-LOC	1
14	be-PER-ORG	1
15	be-MISC-MISC	1
17	be-MISC-PER	1
18	be-LOC-ORG	1
19	be-LOC-PER	1
20	be-MISC-LOC	1
21	be-ORG-MISC	1
22	be-ORG-PER	1
23	be-MISC-ORG	1
24	appoint-ORG-PER	1
25	appoint-PER-ORG	1
27	appoint-PER-LOC	1
28	appoint-MISC-PER	1
30	established-ORG-ORG	1
31	established-PER-LOC	1
32	established-MISC-LOC	1
33	established-ORG-MISC	1
34	established-LOC-ORG	1
36	established-MISC-MISC	1
37	established-MISC-ORG	1
...	...	...
4517	reside-ORG-LOC	1
4518	promulgate-LOC-MISC	1
4519	notify-PER-ORG	1
4521	purify-PER-LOC	1
4523	prescribed-LOC-MISC	1
4527	orchestrate-PER-LOC	1

- cluster 3, 417 typed predicates
-

	data_index	cluster
0	star-PER-MISC	2
10	be-PER-MISC	2
29	appoint-PER-MISC	2
38	established-PER-MISC	2
52	carry-PER-MISC	2
65	give-PER-MISC	2
79	hold-PER-MISC	2
85	express-PER-MISC	2
95	have-PER-MISC	2
105	consist-PER-MISC	2
116	play-PER-MISC	2
138	remain-PER-MISC	2
153	lose-PER-MISC	2
174	establish-PER-MISC	2
176	leave-PER-MISC	2
194	incorporate-PER-MISC	2
202	employ-PER-MISC	2
214	lead-PER-MISC	2
226	write-PER-MISC	2
248	serve-PER-MISC	2
262	bear-PER-MISC	2
265	introduce-PER-MISC	2
273	condemn-PER-MISC	2
283	oppose-PER-MISC	2
286	repeal-PER-MISC	2
287	deliver-PER-MISC	2
302	follow-PER-MISC	2
310	offer-PER-MISC	2
333	manage-PER-MISC	2
337	cause-PER-MISC	2
...	...	...
4384	superimpose-PER-MISC	2
4385	subjugate-PER-MISC	2
4392	fax-PER-MISC	2
4397	frontline-PER-MISC	2
4400	manifest-PER-MISC	2

- cluster 4, 665 typed predicates
-

1	star-PER-PER
13	be-PER-PER
26	appoint-PER-PER
42	established-PER-PER
69	give-PER-PER
81	hold-PER-PER
90	have-PER-PER
108	consist-PER-PER
115	play-PER-PER
139	remain-PER-PER
148	lose-PER-PER
158	beat-PER-PER
166	clash-PER-PER
180	leave-PER-PER
191	bury-PER-PER
193	incorporate-PER-PER
208	employ-PER-PER
221	lead-PER-PER
227	write-PER-PER
245	serve-PER-PER
257	bear-PER-PER
264	introduce-PER-PER
281	oppose-PER-PER
288	deliver-PER-PER
295	follow-PER-PER
308	offer-PER-PER
323	separate-PER-PER
339	cause-PER-PER
343	link-PER-PER
353	become-PER-PER
...	...
4410	narrate-PER-PER
4411	grab-PER-PER
4412	shower-PER-PER
4421	alert-PER-PER
4423	fare-PER-PER
4424	protected-PER-PER

Semantic Parsing Using Relation Clusters

- binary predicates as *packed predicates*: functions from argument types to relations.
- $born_{arg0:PER,argIn:LOC} \rightarrow rel49(\text{"}birthplace\text{"})$ and $born_{arg0:PER,argIn:DAT} \rightarrow rel53(\text{"}birthdate\text{"})$
- $$born \vdash (S\backslash NP)/PP[in] : \lambda y \lambda x. \begin{Bmatrix} (x: PER, y: LOC) \Rightarrow rel49 \\ (x: PER, y: DAT) \Rightarrow rel53 \end{Bmatrix} (x,y)$$
- $born_{arg0,argIn}$ applied to arguments *Obama* and *Hawaii* with type distributions $(PER = 0.9, LOC = 0.1)$ and $(LOC = 0.7, DAT = 0.3)$, the distribution over relations will be $(rel49 = P(PER) * P(LOC) = 0.63, rel53 = P(PER) * P(DAT) = 0.27, etc)$
- *1961* has type-distribution $(LOC = 0.1, DAT = 0.9)$
- *Obama was bron in Hawaii in 1961*:
- $$\begin{Bmatrix} rel49{=}0.63 \\ rel53{=}0.27 \\ \ldots \end{Bmatrix} (ob, hw) \wedge \begin{Bmatrix} rel49{=}0.09 \\ rel53{=}0.81 \\ \ldots \end{Bmatrix} (ob, 1961)$$

Problems and Future Work

- Problems
  - the experiment heavily depends on the predicate-argument structure
  - only proper nouns are used (entity types are only applied on the proper nouns)
  - sparsity

Problems and Future Work

- Problems
  - the experiment heavily depends on the predicate-argument structure
  - only proper nouns are used (entity types are only applied on the proper nouns)
  - sparsity
- Future Work
  - build CCG parser integrated predicate-argument (other structures as well) extractor
  - implement other clustering method to capture other semantic relations (such as hierarchical clustering to capture hypernym relations)
  - explore resources to improve clustering accuracy (such as WordNet and Ontonets)

References

- Mike Lewis, Mark Steedman, *Combined Distributional and Logical Semantics*, 2013
- Gemma Boleda, Aurélie Herbelot, *Formal Distributional Semantics: Introduction to the Special Case*
- Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, Daisuke Bekki. *ccg2lambda: A Compositional Semantics System*. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics - System Demonstrations, pages 85–90, Berlin, Germany, August 7-12, 2016