Combined Distributional and Logical Semantics

Mike Lewis and Mark Steedman
ACL 2013

Presented by : Rahul Jha

Outline

- Motivation
- Background
- Proposed Method
- Experiments

MOTIVATION

Motivation

- Distributional semantics
  - Successfully induce the meaning of content words
  - Not clear how to apply to semantic operators such as determiners, negation, conjunctions etc.
- Formal semantics:
  - Model the operators well
  - But show low recall on practical applications

Prior Work

- Supervised data driven approaches
  - Supervised models of formal semantics for specialized languages (GeoQuery, ATIS)
  - Large semantically annotated corpora (Ontonotes, MeaningBank)
    - Typically map words onto senses in ontologies, but do not support inferences like (*author -> wrote)*

Prior Work

- Unsupervised predicate clustering based on arguments
  - Yao et al. (2011) cluster relations between entities based on lexical, syntactic and semantic features
  - Poon and Domingos (2009) recursively cluster fragments of dependency trees based on arguments
    - But computationally expensive
- Vector space models (Mitchell and Lapata, 2008; Coecke et al., 2010; Grefenstette et al., 2011)
  - Not obvious how to represent logical relations such as quantification in vector spaces
- Natural Logic (MacCartney and Manning, 2007)
  - Can only make inferences b/w two sentences + sensitive to word order
- Parsing models based on formal compositional semantics (Boxer, XLE)

Goals of present work

- To map natural language to first-order logic representations that:
  - Capture the meaning of function words such as *every, not* and *or*
  - Use distributional semantics to model the meaning of content words

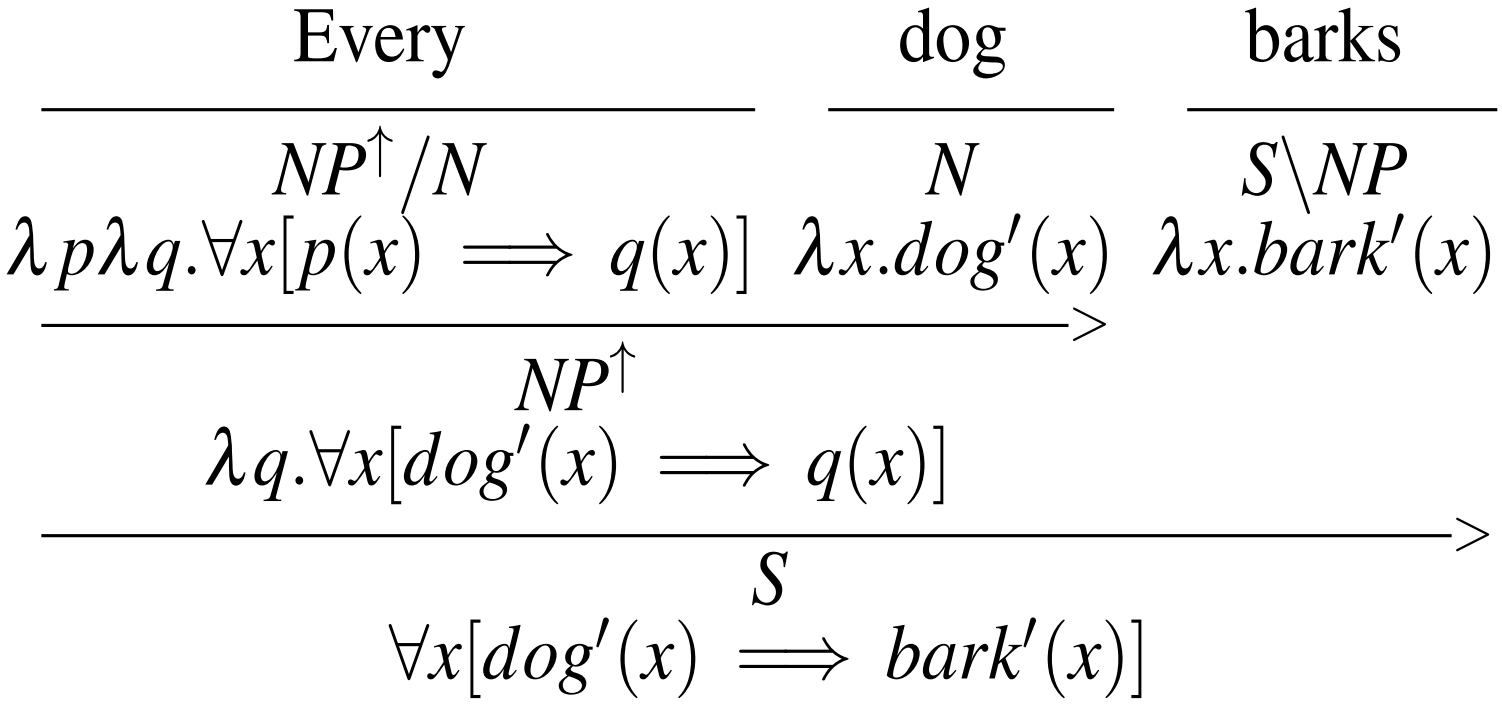
BACKGROUND

CCG

- A lexicalized theory of language where the lexicon encodes:
  - Syntactic category that determines which other categories the word may combine with
  - Semantic interpretation defining the compositional semantics
- Example lexicon entry:

$$write \vdash (S\backslash NP)/NP : \lambda y \lambda x.write'(x,y)$$

Sample CCG Derivation



Limitations of CCG

- Does not generalize over similar predicates, e.g.: *write* and *author* might map to the same predicate
- The goal of the paper is to learn a CCG lexicon that maps equivalent such as these to the same form:

$$author \vdash N/PP[of] : \lambda x \lambda y.relation37(x,y)$$
$$write \vdash (S\backslash NP)/NP : \lambda x \lambda y.relation37(x,y)$$

Typing Predicates

- Assigning allowed types to predicates, e.g., writing is a relation between people and books
- Helps model ambiguous predicates e.g.
  - Born(Obama, Hawaii)
  - Born(Obama, 1961)
- Also improves the clustering

PROPOSED METHOD

Step 1: Initial Semantic Analysis

- Use the C&C parser (Clark and Curran, 2004) to create a CCG parse
- Semantic lexical entries for most words can be generated automatically from syntactic category and the POS tag
- For some function words, manual lexical entries are created

Step 2: Induce entity types

- Uses an LDA style model
- Constructs a pseudo-document for each unary predicate (e.g. $born_{arg1}$) based on all its argument entities
- Entity types are topics learnt by LDA in the above pseudo-document set
- Each type has a multinomial distribution over arguments [e.g *P(Hawalii|**LOC**)* > *P(1961|**LOC**)*]
- Each predicate has a multinomial distribution over topics [*e.g P(LOC|**born**); P(DAT|**born**) > P(SONG|**born**)*]

Inferring argument type during parsing

- An initial type distribution is applied based on the learned topics
- The probabilities are updated during parsing:

$$\dfrac{\text{file} \qquad\qquad \text{a suit}}{\dfrac{(S\backslash NP)/NP \qquad\qquad NP^{\uparrow}}{\lambda y{:}\left\{\begin{array}{c} DOC=0.5 \\ LEGAL=0.4 \\ CLOTHES=0.01 \\ \cdots \end{array}\right\} \lambda x{:}\left\{\begin{array}{c} PER=0.7 \\ ORG=0.2 \\ \cdots \end{array}\right\}.file_{arg0,arg1}(x,y) \quad \lambda p.\exists y{:}\left\{\begin{array}{c} CLOTHES=0.6 \\ LEGAL=0.3 \\ DOC=0.001 \\ \cdots \end{array}\right\}[suit'(y)\wedge p(y)]}{S\backslash NP}<$$

$$\lambda x{:}\left\{\begin{array}{c} PER=0.7 \\ ORG=0.2 \\ \cdots \end{array}\right\} \exists y{:}\left\{\begin{array}{c} LEGAL=0.94 \\ CLOTHES=0.05 \\ DOC=0.004 \\ \cdots \end{array}\right\})[suit'(y)\wedge file_{arg0,arg1}(x,y)]$$

Step 3: Cluster the predicates

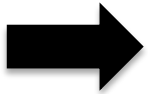
- Parse all sentences as above and obtain word count vectors for each typed predicate
- Cluster these vectors using the Chinese Whispers algorithm

Parsing New Sentences

- Since the predicates are ambiguous between types before complete parsing is done, the parser produces packed representations

$$born \vdash (S\backslash NP)/PP[in] :$$
$$\lambda y \lambda x. \left\{ \begin{array}{l} (x: PER, y: LOC) \Rightarrow rel49 \\ (x: PER, y: DAT) \Rightarrow rel53 \end{array} \right\} (x, y)$$

Obama was born in Hawaii in 1961



$$\left\{ \begin{array}{c} rel49{=}0.63 \\ rel53{=}0.27 \\ \ldots \end{array} \right\} (ob, hw) \wedge \left\{ \begin{array}{c} rel49{=}0.09 \\ rel53{=}0.81 \\ \ldots \end{array} \right\} (ob, 1961)$$

EXPERIMENTS

Experiments: Question Answering

- Generate sample questions from New York Times subset of Gigaword from 2010
- Search for answers in 2009

System	Answers	Correct
Relational-LDA	7046	11.6%
Reverb	180	89.4%
CCG-Baseline	203	95.8%
CCG-WordNet	211	94.8%
CCG-Distributional@250	250	94.1%
CCG-Distributional@500	500	82.0%

Experiments: Textual Entailment

- Given a premise and a hypothesis, return Yes, No or Unknown for whether the hypothesis follows from the premise

System	Single Premise	Multiple Premises
MacCartney&Manning 07	84%	-
MacCartney&Manning 08	98%	-
CCG-Dist (parser syntax)	70%	50%
CCG-Dist (gold syntax)	89%	80%

Discussion

- Do the experimental results justify the model?
- Applicable to specialized corpora?
- Applicable to higher level problems like relation extraction, summarization etc.?