Predicting Rich Linguistic Structure with Neural Networks

Jan Buys
University of Oxford

OXFORD
N L P

Goals of Natural Language Processing

Natural Language Understanding

- Tasks that require inference, reasoning
- What representations do we need?
- How much structure?
- Where should this structure come from?

Goals of Natural Language Processing

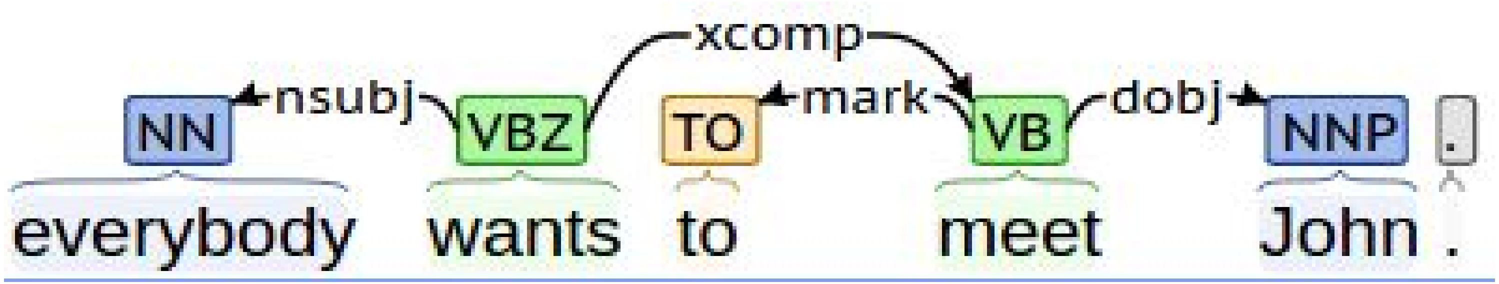
How to get there:

- Powerful models with neural networks
- Linguistically-informed structure
- Accurate inference over simple underlying structure

Robust Incremental Neural Semantic Graph Parsing

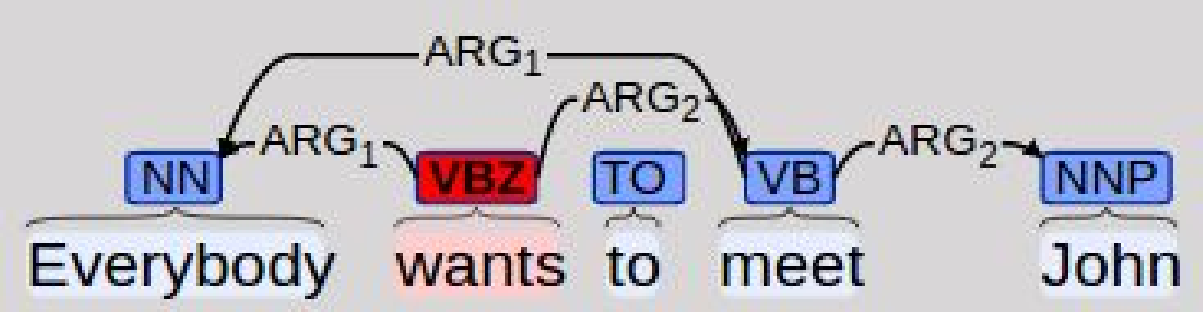
Jan Buys and Phil Blunsom, ACL 2017

Syntactic Dependencies



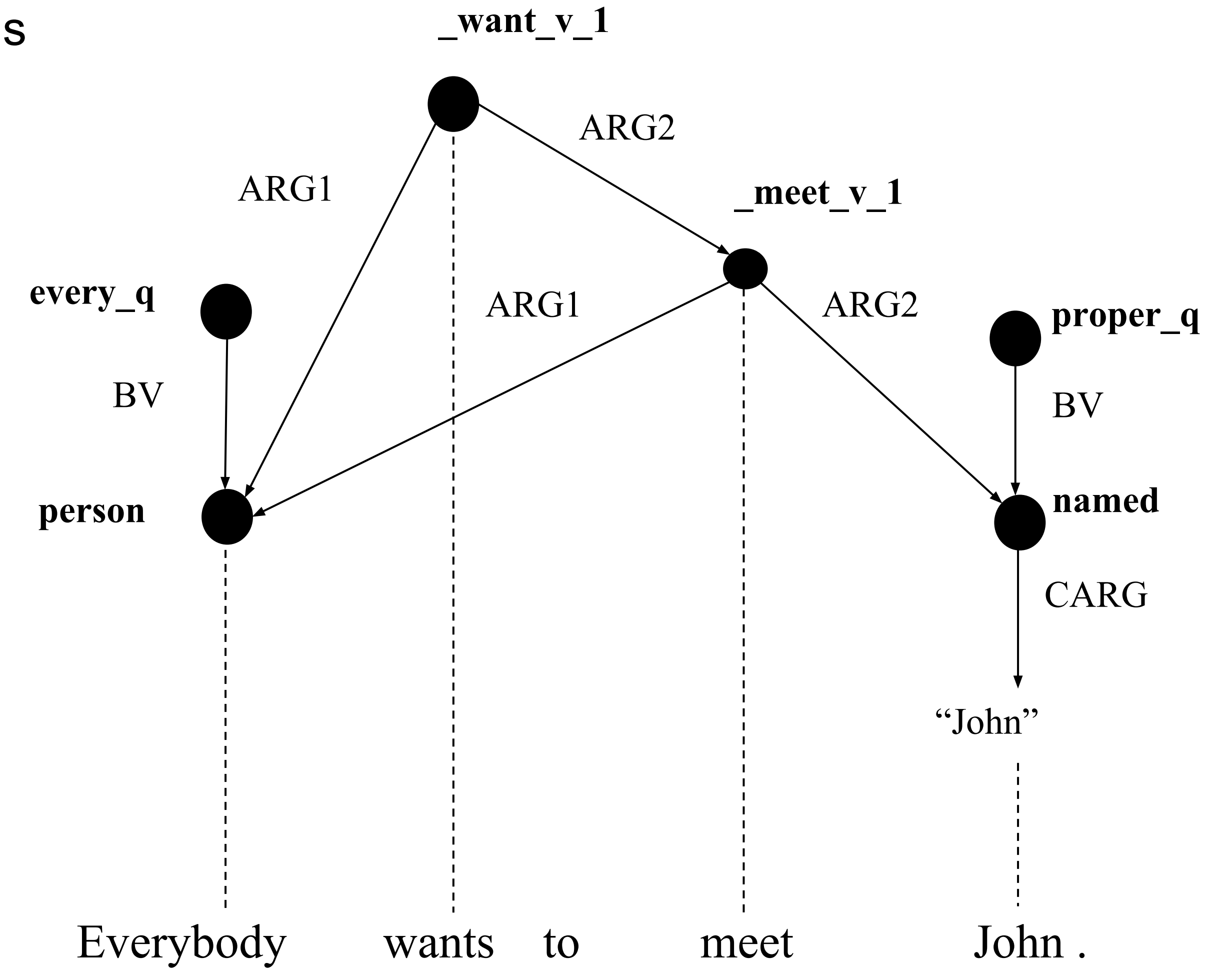
(Dyer et al 2015; Andor et al, 2016; Kiperwasser and Goldberg 2016)

Semantic Dependencies

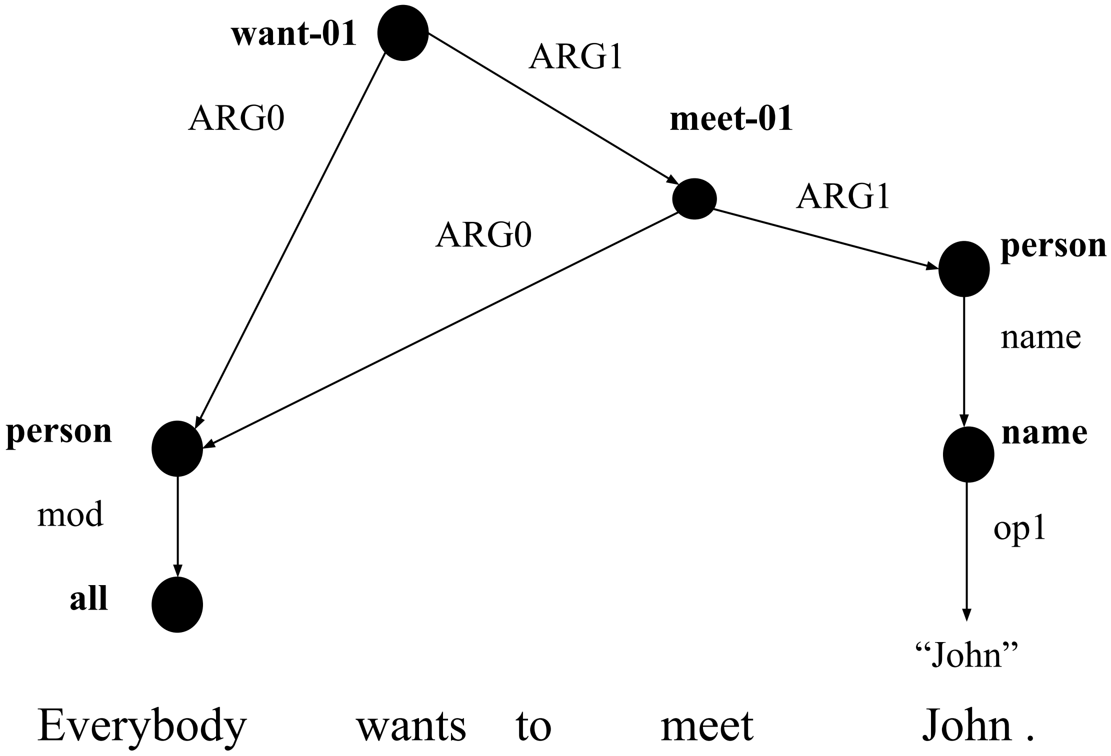


(Oepen et al, 2014, 2015; Martins and Almeida 2014)

Semantic Graphs



Abstract Meaning Representation



(Banarescu et al, 2013; Flanigan et al, 2014; Wang et al, 2015; Arzi et al, 2015)

Semantic representations

Compositionality

- Semantics constrained by syntax
- Explicit alignment to sentence structure
- Underlying lexicon and grammar

Semantic representations

Compositionality

- Comprehensiveness
- Consistency
- Scalability

(Bender et al., 2015)

Semantic representations

Minimal Recursion Semantics

- Semantic representation for feature-structure formalisms such as HPSG
- Implemented in the English Resource Grammar (ERG)
- Treebanks based on the ERG: DeepBank (WSJ), Redwoods.

(Copenstake et al., 2005; Flickinger 2000; Flickinger et al., 2017)

Parsing MRS

Existing parsers based on the ERG

- High precision, but incomplete coverage
- Find HPSG derivations
- Obtain MRS through unification
- Score with MaxEnt model

TOP h1
INDEX e3

RELS {

person(0:9)	
LBL	*h4*
ARG0	*x5*

every_q(0:9)	
LBL	*h6*
ARG0	*x5*
RSTR	*h7*
BODY	*h8*

_want_v_1(10:15)	
LBL	*h2*
ARG0	*e3*
ARG1	*x5*
ARG2	*h9*

_meet_v_1(19:23)	
LBL	*h10*
ARG0	*e11*
ARG1	*x5*
ARG2	*x12*

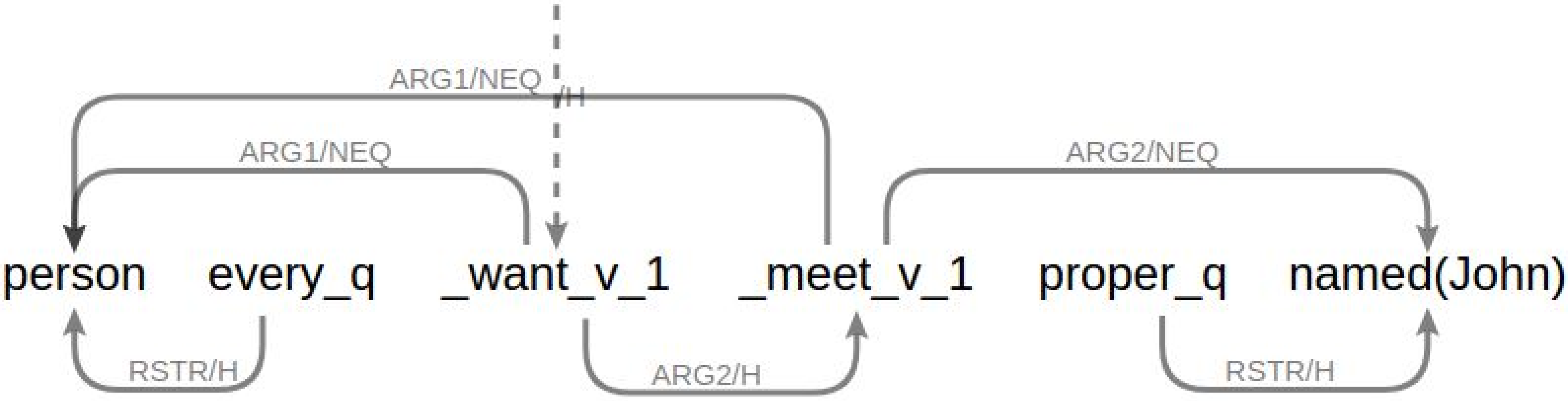
proper_q(24:28)	
LBL	*h13*
ARG0	*x12*
RSTR	*h14*
BODY	*h15*

named(24:28)	
LBL	*h16*
ARG0	*x12*
CARG	*John*

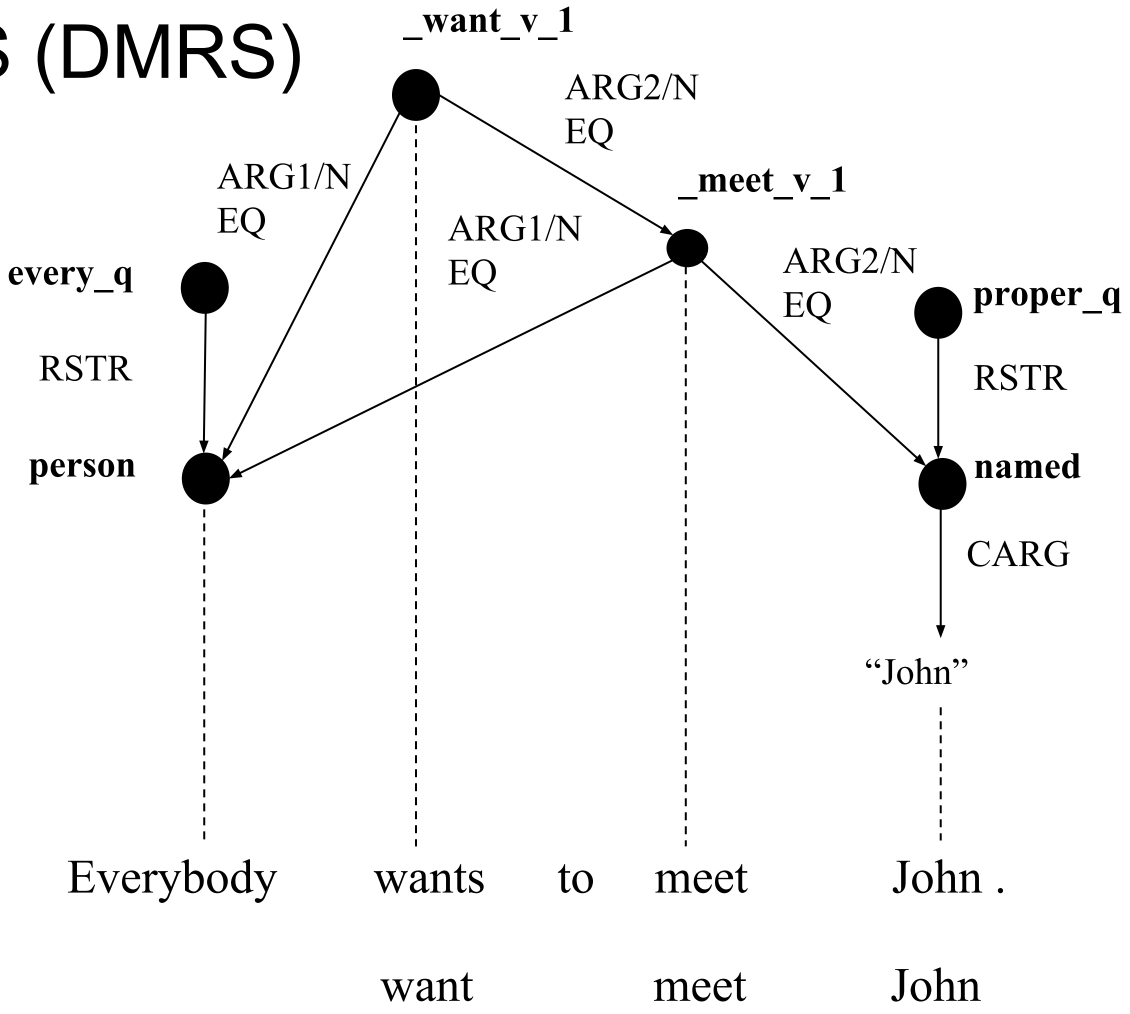
}

HCONS { h1 =q h2, h7 =q h4, h9 =q h10, h14 =q h16 }

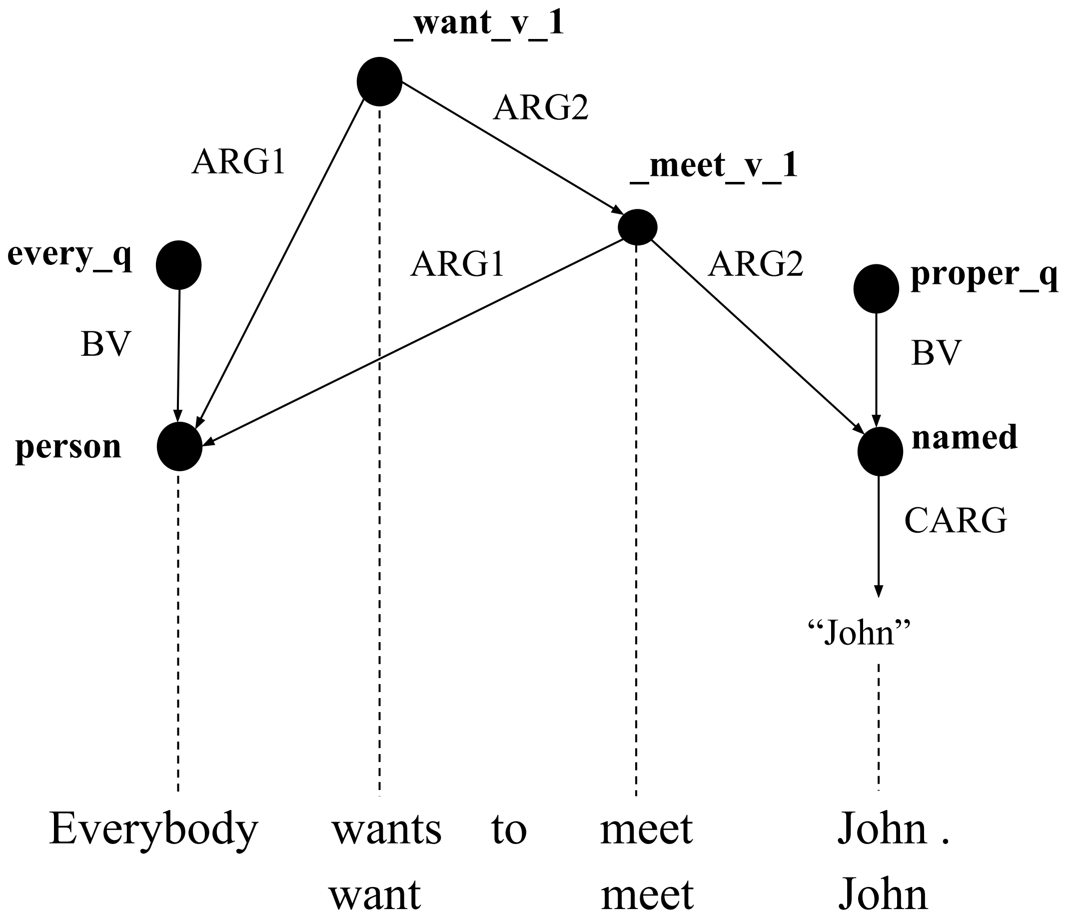
Dependency MRS (DMRS)



Dependency MRS (DMRS)



Elementary Dependency Structure (EDS)



End-to-end semantic graph parsing

- Text to directed acyclic graphs with labelled nodes and edges, each node aligned to a span of input tokens
- No intermediate syntactic structure

End-to-end semantic graph parsing

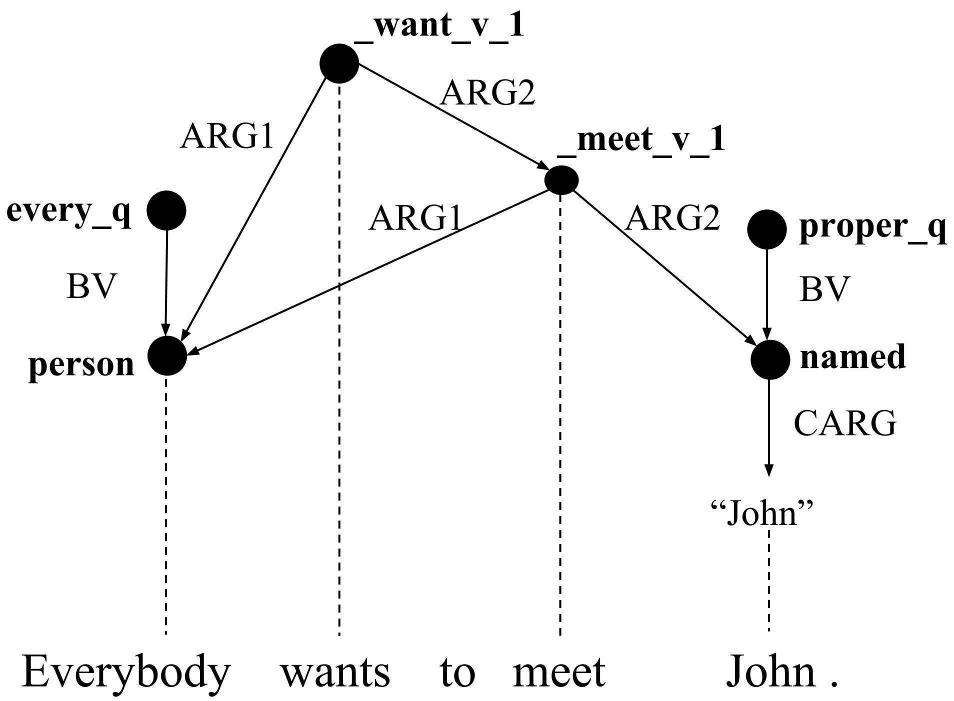
Predict graph structure incrementally

- Top-down linearization
- Transition-based parsing
  - Generating nodes rather than using words as nodes

End-to-end semantic graph parsing

Top-down graph linearization

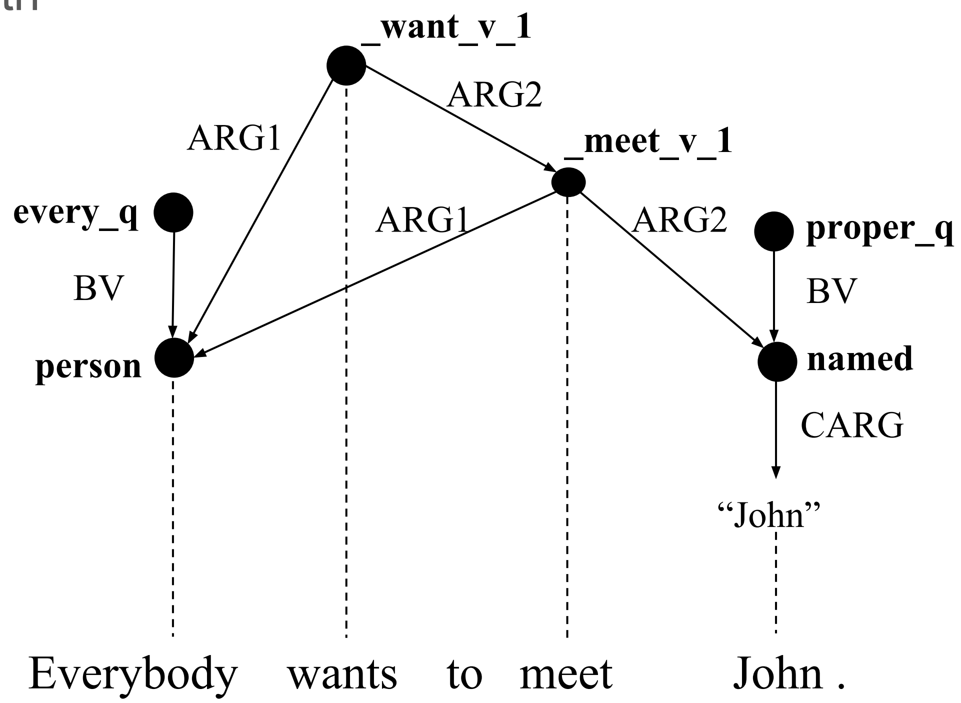
```
:root(  _want_v_1
  :ARG1( person
    :BV-of( every_q ) )
  :ARG2 _meet_v_1
    :ARG1*( person
    :ARG2( named_CARG
      :BV-of ( proper_q ) ) )
```



End-to-end semantic graph parsing

Top-down graph linearization with alignments

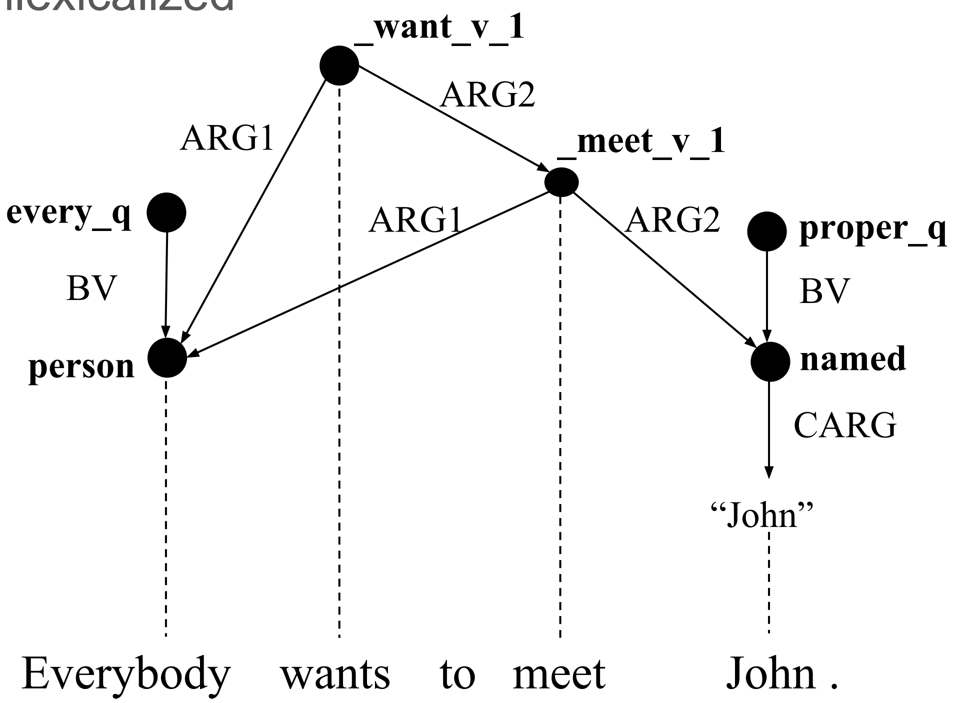
```
:root( <2> _want_v_1
  :ARG1( <1> person
    :BV-of( <1> every_q ) )
  :ARG2 <4> _meet_v_1
    :ARG1*( <1> person
    :ARG2( <5> named_CARG
      :BV-of ( <5> proper_q ) ) )
```



End-to-end semantic graph parsing

Top-down graph linearization, unlexicalized

```
:root( <2> _v_1
  :ARG1( <1> person
    :BV-of( <1> every_q ) )
  :ARG2 <4> _v_1
    :ARG1*( <1> person
    :ARG2( <5> named_CARG
      :BV-of ( <5> proper_q ) ) )
```



Transition-based graph parsing

- Arc-eager transition system for semantic graphs
- Data structures: Input sentence, stack, buffer
- Actions:
  - Shift - generate next predicate on buffer
  - Reduce
  - Left-arc
  - Right-arc
  - Cross-arc

Transition-based parsing

Everybody wants to meet John .

Transition	Stack	Buffer
Init(1, person)		(1, person)

Transition-based parsing

Everybody wants to meet John .

Transition	Stack	Buffer
Shift(1, every_q)	(1, person)	(1, every_q)

Transition-based parsing

Everybody wants to meet John .

Transition	Stack	Buffer
Left-arc(BV)	(1, person)	(1, every_q)

Transition-based parsing

Everybody **wants** to meet John .

Transition	Stack	Buffer
Shift(2, _v_1)	(1, person), (1, every_q)	(2, _want_v_1)

Transition-based parsing

Everybody **wants** to meet John .

Transition	Stack	Buffer
Reduce	(1, person)	(2, _want_v_1)

Transition-based parsing

Everybody **wants** to meet John .

Transition	Stack	Buffer
Left-arc(ARG1)	(1, person)	(2, _want_v_1)

Transition-based graph parsing

Transition-based parsing: Oracle

- Node ordering - monotone ordering based on alignments
- Predict alignment spans start (shift) and end (reduce)

End-to-end graph parsing: Encoder-decoders

Formulate parsing as a sequence to sequence problem

- Use either top-down or transition-based linearization

RNN Encoder-decoders

- LSTM encodes the sentence (based on word embeddings)
- A decoder LSTM then decodes the sequence
- Attention mechanism links the encoder and decoder
  - Soft alignment between input and output learned jointly with the model

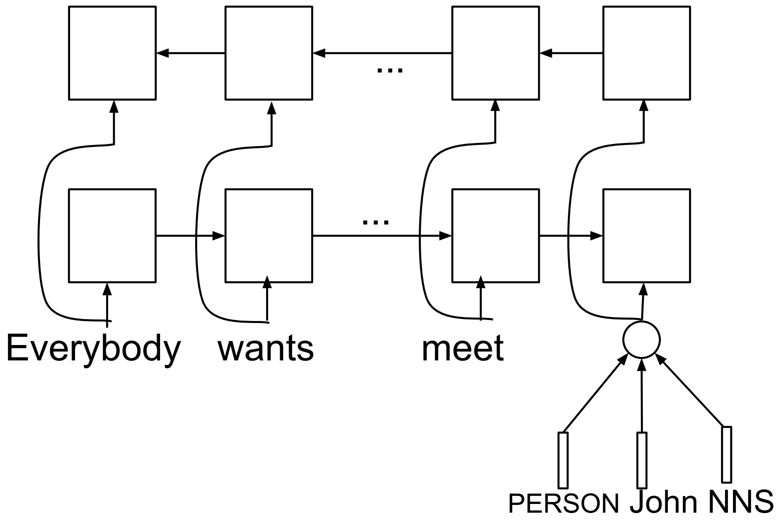
Graph parsing with stack-based encoder-decoders

Alternative to soft attention

- Use alignments of top stack and buffer nodes to extract features based on biLSTM encoder
- Decoder still encodes predicted output symbols with an RNN

Graph parsing with stack-based encoder-decoders

Bidirectional RNN encoder



Graph parsing with stack-based encoder-decoders

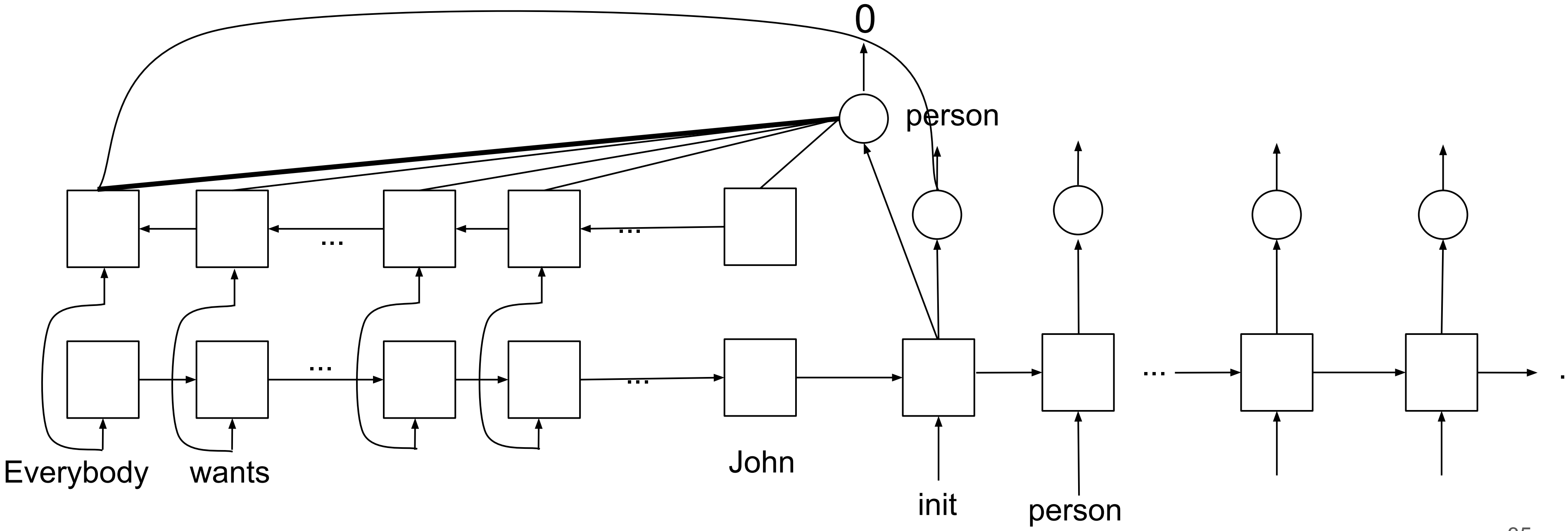
RNN decoder with hard attention

Input sentence **e**, transition sequence **t**, alignment **a**.

$$p(\mathbf{t}, \mathbf{a}|\mathbf{e}) = \prod_{j=1}^{J} p(a_j|(\mathbf{a}, \mathbf{t})_{1:j-1}, \mathbf{e})p(t_j|\mathbf{a}_{1:j}, \mathbf{t}_{1:j-1}, \mathbf{e}).$$

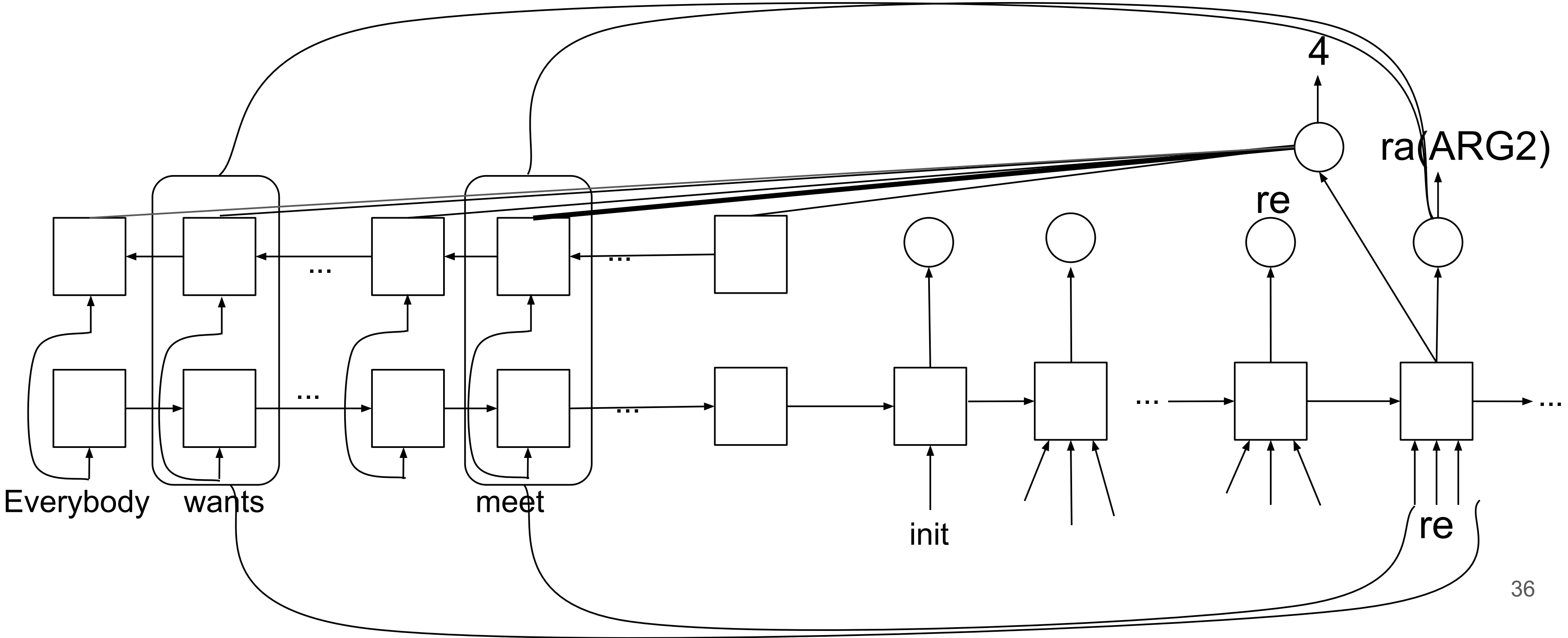
Graph parsing with stack-based encoder-decoders

RNN decoder with hard attention



Graph parsing with stack-based encoder-decoders

RNN decoder with stack-based features



DMRS Experiments

Encoder-decoders with pointer networks for alignment

Model	**EDM**	**EDM Predicates**	**EDM Arguments**
Top-down soft att	81.53	85.32	76.94
Top-down hard att	82.75	86.37	78.37
Arc-eager soft att, lexicalized	81.35	85.79	76.02
Arc-eager soft att, unlexicalized	82.56	86.76	77.54
Arc-eager hard att	84.65	87.77	80.85
Arc-eager stack-based att	**85.28**	**88.38**	**81.51**

DMRS Experiments

Test results

Model	**RNN Top-down**	**RNN Arc-eager**	**ACE (ERG)**
EDM	79.68	**84.16**	**89.64**
EDM Predicates	83.36	87.54	92.08
EDM Arguments	75.16	80.10	86.77
Smatch	85.28	86.69	93.50

DMRS Experiments

Parsing speed

Model	Tokens per second
ACE	7.47
AE RNN	41.63
AE RNN (batched)	529.42

AMR Parsing

- Structure AMR to look more like MRS graphs
- Automatic word alignments
- Classify concepts as surface or abstract

AMR Experiments

Test results

Model	**Smatch**
JAMR - Flanigan et al. (2014)	56
CAMR - Wang et al. (2016)	66.54
ArcEager NN - Damonte et al. (2017)	64
Neural - Peng et al. (2017)	52
Neural - Barzdins and Gosko (2016)	43.3
Top-down attention	56.56
Arc-eager stack-based attention, unlexicalized	60.11

Future work: Semantic graph parsing

- More parsers for MRS
- Semi-supervised learning
- Downstream applications
- Generation

Conclusion

- Neural Networks can effectively predict structured representations
- Robust parser for linguistically sound and informative semantic graphs