Statistical Script Learning with Multi-Argument Events

Pichotta & Mooney, 2014

Esther van den Berg
Professor Manfred Pinkal
Selected Topics in Semantics and Discourse
(WS 2015/16)

UNIVERSITÄT DES SAARLANDES

Topic is still:

Statistical Script Learning with Multi-Argument Events

Pichotta & Mooney, 2014

Esther van den Berg
Professor Manfred Pinkal
Selected Topics in Semantics and Discourse
(WS 2015/16)

Topic is still:

Statistical Script Learning with Multi-Argument Events

Pichotta & Mooney, 2014

As opposed to verb-dependent pair

Topic is still:

Statistical Script Learning with Multi-Argument Events

Pichotta & Mooney, 2014

As opposed to verb-dependent pair

State of the art

Aim:

Modeling interactions between entities in a script

Aim: Modeling interactions between entities in a script

Method: Statistical modeling

Aim: Modeling interactions between entities in a script

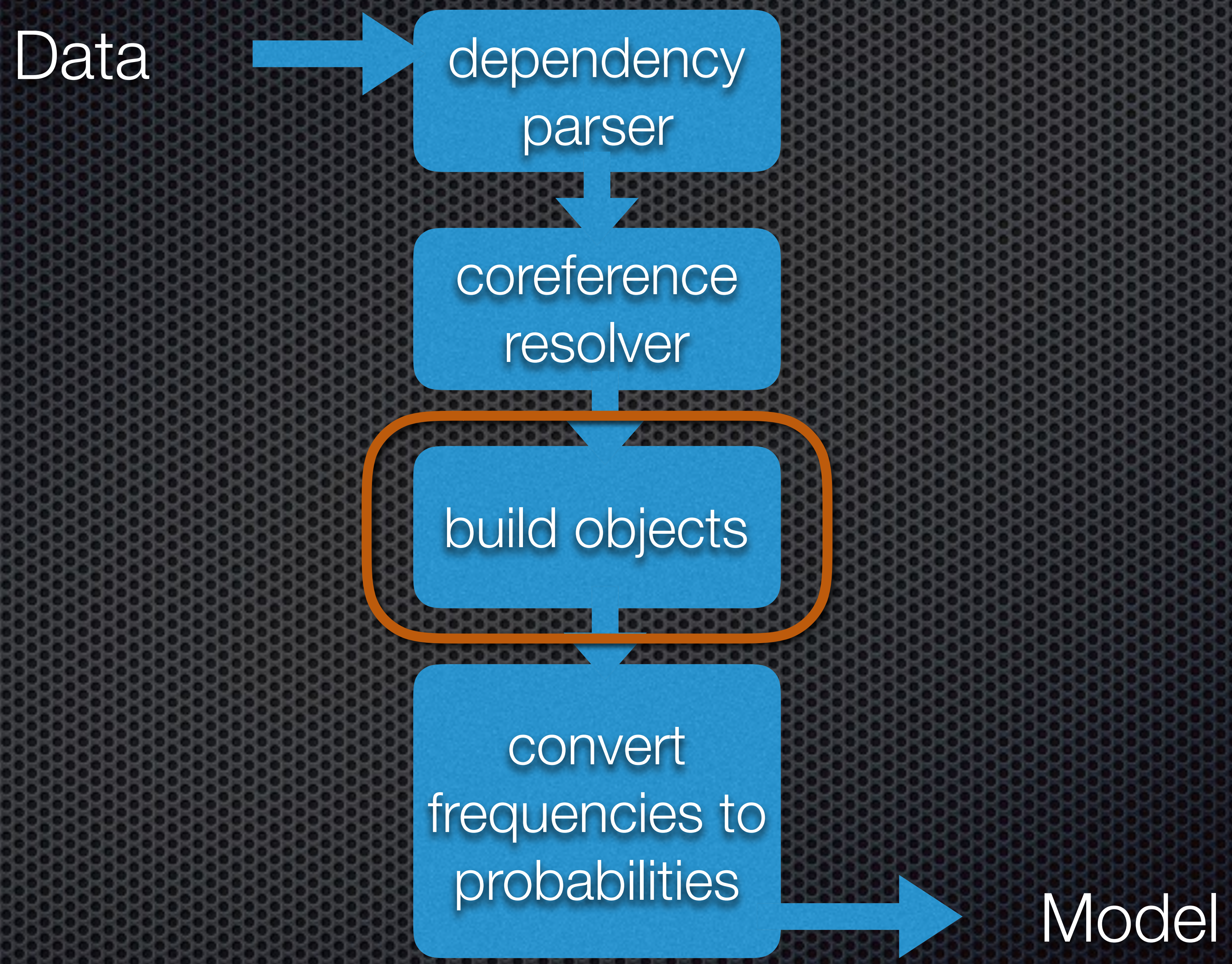
Method: Statistical modeling

Evaluation: Narrative cloze test

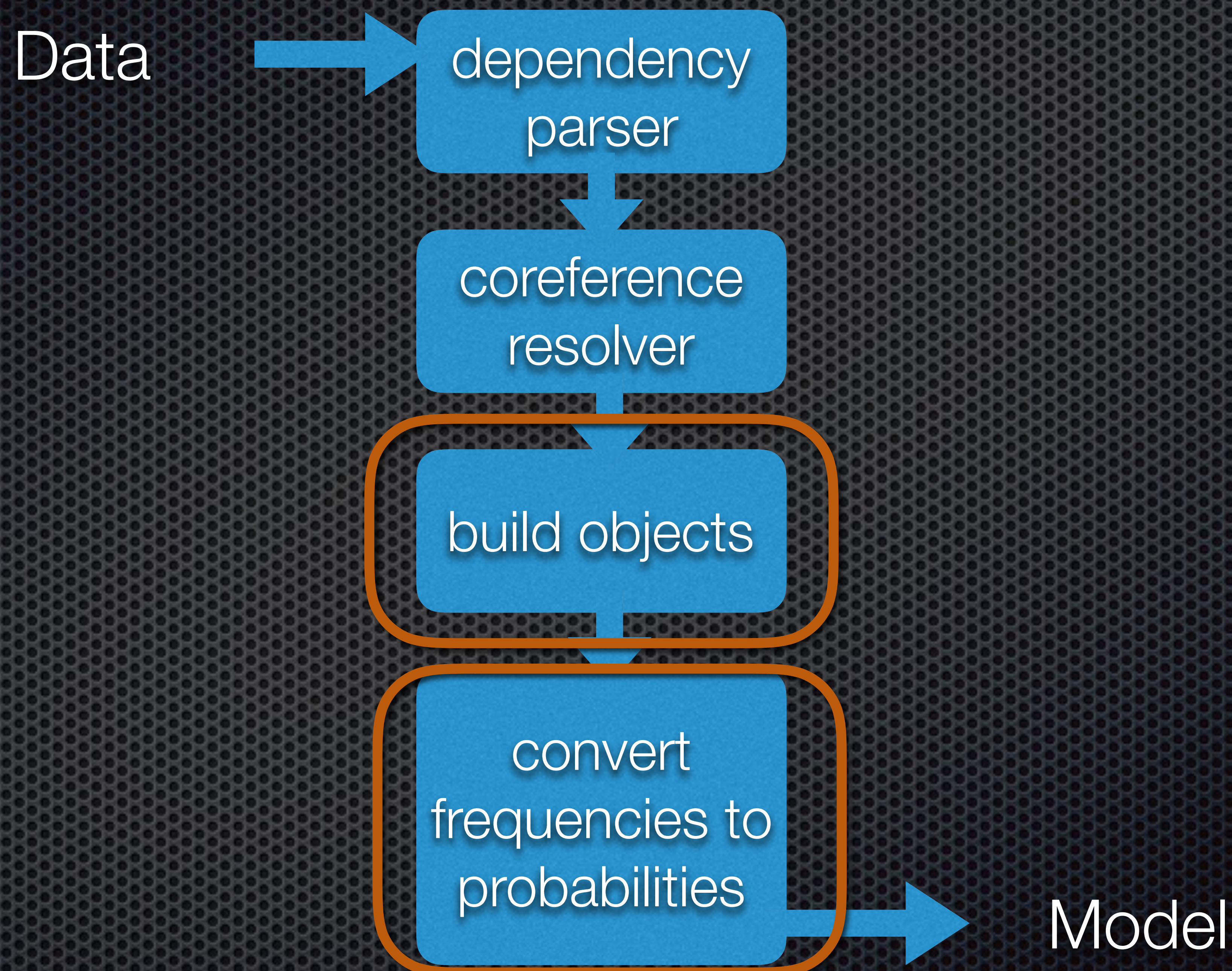
Compared to previous work

- Mooney and DeJong (1985)
  - non-statistical
- Chambers and Jurafsky (2008) / Jans et al. (2012)
  - only models action of single participant
- Chambers and Jurafsky (2009)
  - extension to multi-participant case

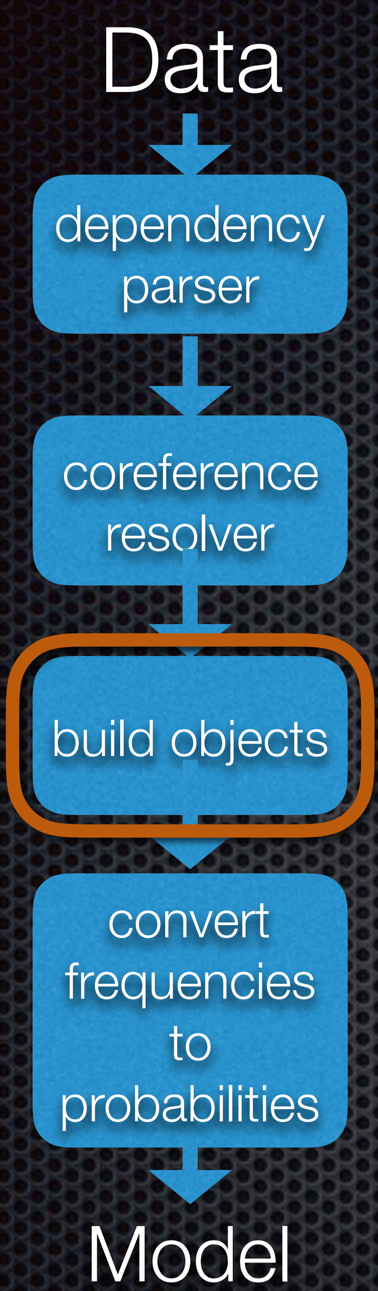
Compared to previous work



Compared to previous work



Pairs


Data
dependency parser
coreference resolver
build objects
convert frequencies to probabilities
Model

Pairs

Mary e-mailed Jim and he responded to her immediately

2 separate chains:

(*email*, *subject*)
(*respond*, *object*)

(*email*, *object*)
(*respond*, *subject*)

Multiargument events

Mary e-mailed Jim and he responded to her immediately

$$v(e_s,\ e_o,\ e_p)$$

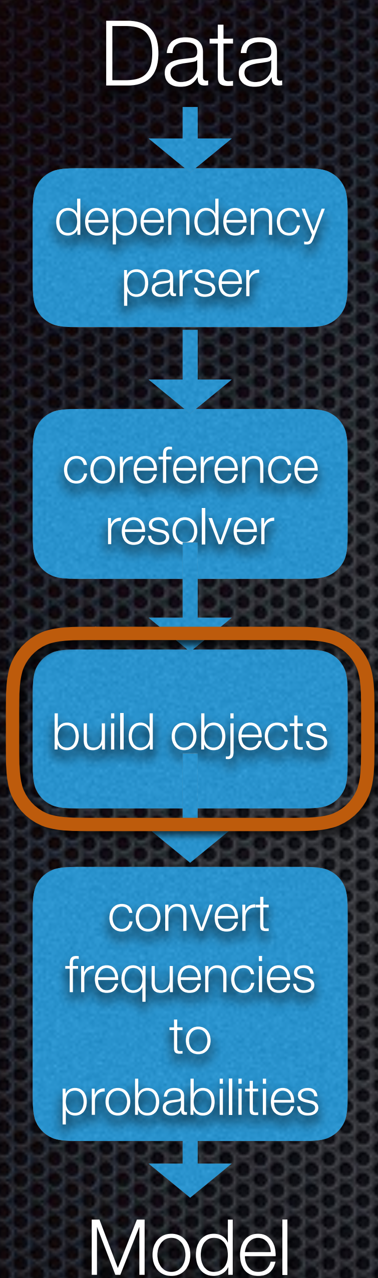
v = verb lemma

e_s = subject

e_o = direct object

e_p= prepositional relation

Multiargument events



Mary e-mailed Jim and he responded to her immediately

$$v(e_s,\ e_o,\ e_p) \quad ➡ \quad \text{e-mail(Mary, Jim,•)}$$

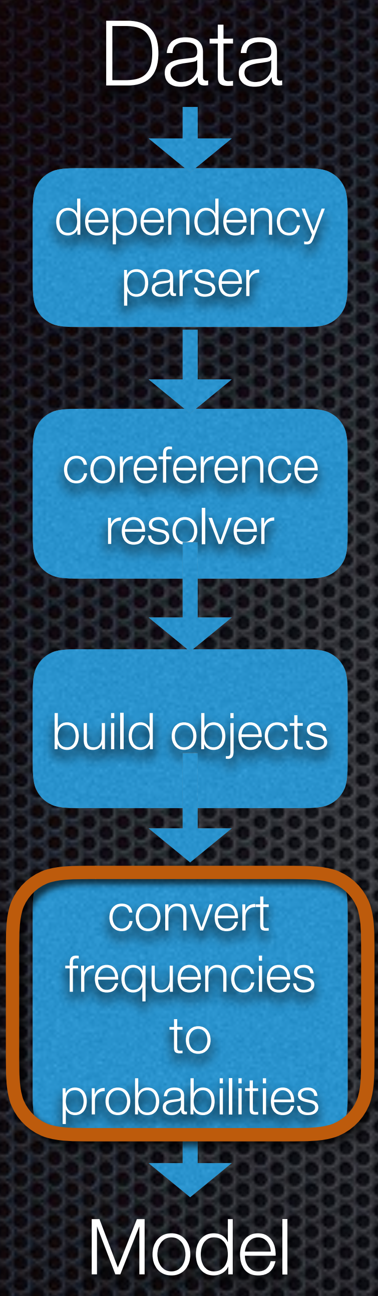
v = verb lemma

e_s = subject

e_o = direct object

e_p= prepositional relation

Overlap with previous work

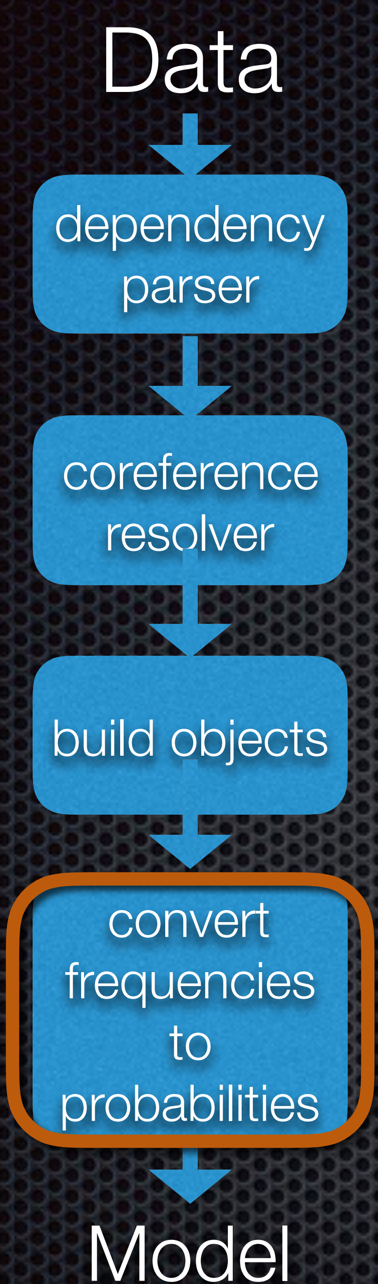


$$P(a_2 \mid a_1) = \frac{c(a_1, a_2)}{c(a_1)}$$

- Old method:
  - traverse corpus, count the number of times each pair co-occurs

problematic for some reason?

Overlap with previous work



$v(e_s,\ e_o,\ e_p)$ get mapped to $\{x,\ y,\ x,\ O\}$

Proposed method

Algorithm 1 Learning with entity substitution

for $a_1, a_2 \in$ `evs` **do**
    $N(a_1, a_2) \leftarrow 0$
end for
for $D \in$ `documents` **do**
    for $a_1, a_2 \in$ `coocurEvs` (D) **do**
        for $\sigma \in$ `subs` (a_1, a_2) **do**
            $N(\sigma(a_1), \sigma(a_2)) \mathrel{+}= 1$
        end for
    end for
end for

Proposed method

Algorithm 1 Learning with entity substitution

for $\boxed{a_1, a_2 \in \texttt{evs}}$ **do**
  $N(a_1, a_2) \leftarrow 0$
end for
for $D \in \texttt{documents}$ **do**
  for $a_1, a_2 \in \texttt{coocurEvs}(D)$ **do**
    for $\sigma \in \texttt{subs}(a_1, a_2)$ **do**
      $N(\sigma(a_1), \sigma(a_2)) \mathrel{+}= 1$
    end for
  end for
end for

set of all
$v(e_s,\ e_o,\ e_p)$

Proposed method

Algorithm 1 Learning with entity substitution

for $a_1, a_2 \in$ `evs` **do**
  $N(a_1, a_2) \leftarrow 0$
end for
for $D \in$ `documents` **do**
  for $a_1, a_2 \in$ `coocurEvs` (D) **do**
    for $\sigma \in$ `subs` (a_1, a_2) **do**
      $N(\sigma(a_1), \sigma(a_2)) \mathrel{+}= 1$
    end for
  end for
end for

co-occurence matrix

Proposed method

returns ordered pairs of co-occuring events in D

Algorithm 1 Learning with entity substitution

```
for a1, a2 ∈ evs do
    N(a1, a2) ← 0
end for
for D ∈ documents do
    for a1, a2 ∈ coocurEvs(D) do
        for σ ∈ subs(a1, a2) do
            N(σ(a1), σ(a2)) += 1
        end for
    end for
end for
```

Proposed method

Algorithm 1 Learning with entity substitution

for $a_1, a_2 \in$ `evs` **do**
    $N(a_1, a_2) \leftarrow 0$
end for
for $D \in$ `documents` **do**
    for $a_1, a_2 \in$ `coocurEvs` (D) **do**
        for $\sigma \in \boxed{\texttt{subs}\,(a_1, a_2)}$ **do**
            $N(\sigma(a_1), \sigma(a_2)) \mathrel{+}= 1$
        end for
    end for
end for

mapping to
$v(e_s,\ e_o,\ e_p)$

Proposed method

returns ordered pairs of co-occuring events in D

mapping to $v(e_s,\ e_o,\ e_p)$

Algorithm 1 Learning with entity substitution

```
for a1, a2 ∈ evs do
    N(a1, a2) ← 0
end for
for D ∈ documents do
    for a1, a2 ∈ coocurEvs(D) do
        for σ ∈ subs(a1, a2) do
            N(σ(a1), σ(a2)) += 1
        end for
    end for
end for
```

set of all $v(e_s,\ e_o,\ e_p)$

co-occurence matrix

Proposed method

Algorithm 1 Learning with entity substitution

for $a_1, a_2 \in$ `evs` **do**
  $N(a_1, a_2) \leftarrow 0$
end for
for $D \in$ `documents` **do**
  for $a_1, a_2 \in$ `coocurEvs` (D) **do**
    for $\sigma \in$ `subs` (a_1, a_2) **do**
      $N(\sigma(a_1), \sigma(a_2))$ += 1
    end for
  end for
end for

$$P(a_1, a_2) = \frac{N(a_1, a_2)}{\sum_{a'_1, a'_2} N(a'_1, a'_2)}.$$

Two evaluation tasks

- predicting pair-events to compare to previous methods
- guessing multi-argument event, given all other events and the entities mentioned in the held-out event
    - entities rewritten to x,y,z such that distinct entities never get rewritten to the same variable
    - entity not in both training and held-out mapped to O

Experimental set-up

- Stanford dependency parser
- Stanford coreference resolution engine
- NYT portion of Gigaword Corpus
  - 1.1 m articles for training
  - 10 000 held-out events
  - 500 disjoint development events

Results

Method	R@10	Accuracy
Random	0.001	0.495
Unigram	0.297	0.552
Single Protagonist	0.282	0.553
Joint Pair	**0.336**	**0.561**

Table 2: Results for pair events.

Results

Method	R@10	Accuracy
Random	0.001	0.334
Unigram	0.216	0.507
Multiple Protagonist	0.209	0.504
Joint	**0.245**	**0.549**

Table 1: Results for multi-argument events.

scores events by summing the scores the single-protagonist model assigns to pairs in the set $v(e_s, e_o, e_p)$

Results

Method	R@10	Accuracy
Random	0.001	0.334
Unigram	0.216	0.507
Multiple Protagonist	0.209	0.504
Joint	**0.245**	**0.549**

Table 1: Results for multi-argument events.

Conclusion

Aim:
Modeling interactions between entities in a script

- State of the art:
  - Statistical learning of multiargument events is possible

Conclusion

Aim:
Modeling interactions between entities in a script

- State of the art:
  - Statistical learning of multiargument events is possible
- Other contributions of the paper:
  - new unigram baseline
  - new ‘joint pair’ baseline
  - new accuracy metric

Conclusion

Aim:
Modeling interactions between entities in a script

- State of the art:
  - Statistical learning of multiargument events is possible
- Other contributions of the paper:
  - new unigram baseline
  - new 'joint pair' baseline
  - new accuracy metric

➡ evaluation methodology for scripts not yet standardised

Questions?

- about the ranking method?

$$S(a) = \sum_{i=1}^{p-1} \log P(a|a_i) + \sum_{i=p}^{|A|} \log P(a_i|a) \quad (5)$$

- about the baselines?

$$M(a) = S_{e_s}((v, \mathrm{subj})) + S_{e_o}((v, \mathrm{obj})) + S_{e_p}((v, \mathrm{prep}))$$

Thank you!