Learning Statistical Scripts with LSTM Recurrent Neural Networks

Karl Pichotta & Raymond J. Mooney
Department of Computer Science
The University of Texas at Austin

AAAI 2016

Motivation

- *Following the Battle of Actium, Octavian invaded Egypt. As he approached Alexandria, Antony's armies deserted to Octavian on August 1, 30 BC.*
    - Did Octavian defeat Antony?

Motivation

- *Following the Battle of Actium, Octavian invaded Egypt. As he approached Alexandria,* ***Antony's armies deserted to Octavian*** *on August 1, 30 BC.*
  - Did Octavian defeat Antony?

Motivation

- *Antony's armies deserted to Octavian*

  ⇒

  Octavian defeated Antony

- Not simply a paraphrase rule!

- Need world knowledge.

Scripts

- **Scripts**: models of events in sequence.
  - "**Event**": verb + arguments.
  - Events don't appear in text randomly, but according to world dynamics.
  - Scripts try to capture these dynamics.
  - Enable automatic inference of implicit events, given events in text (e.g. *Octavian defeated Antony*).

Outline

- Background
- Methods
- Experiments
- Conclusion

Outline

- Background
  - Statistical Scripts
  - Recurrent Neural Nets

Background: Statistical Scripts

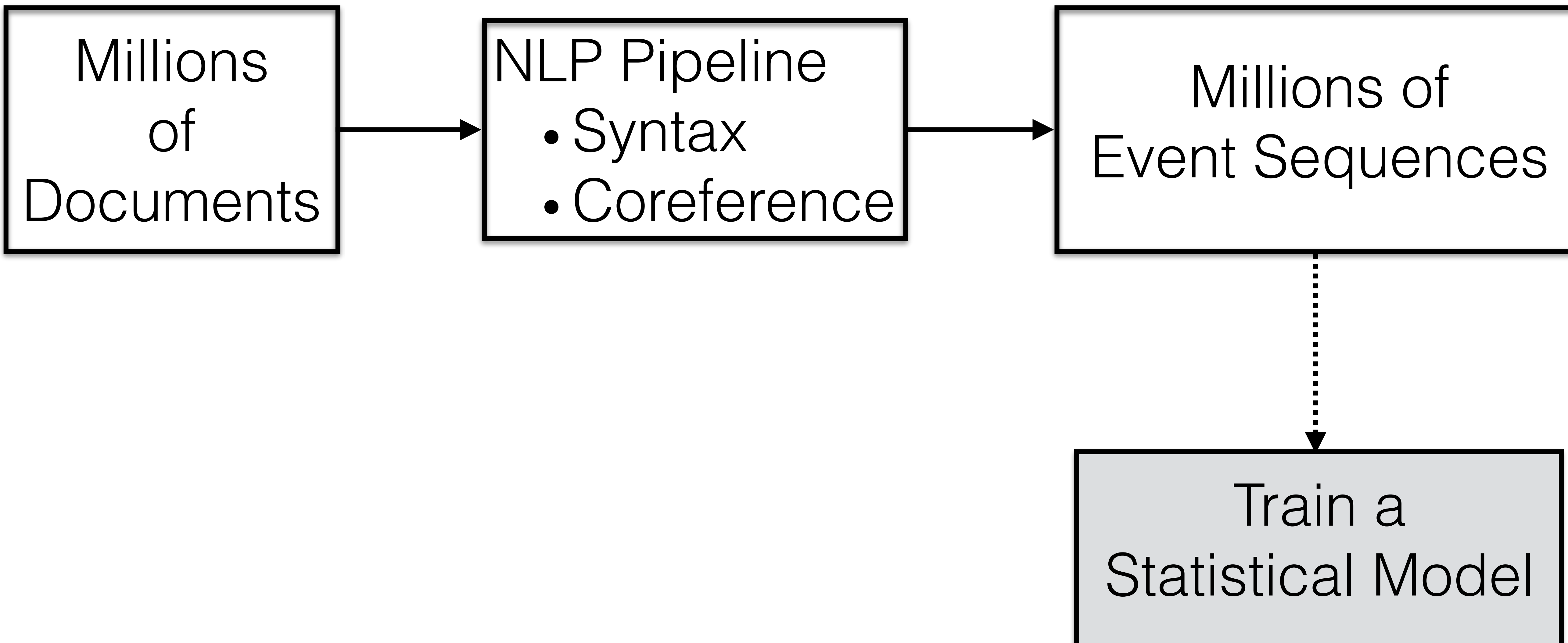
- **Statistical Scripts**: Statistical Models of Event Sequences.

- Non-statistical scripts date back to the 1970s [Schank & Abelson 1977].

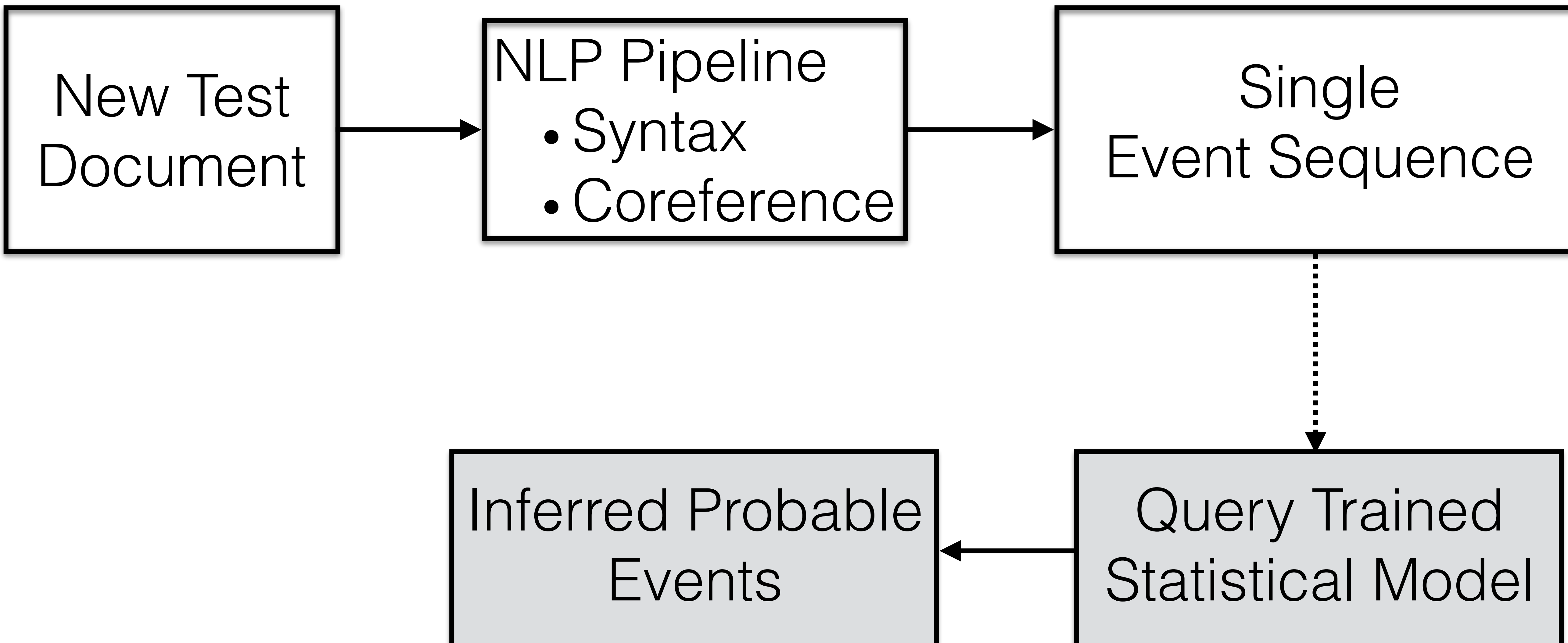
- Statistical script learning is a small-but-growing subcommunity [e.g. Chambers & Jurafsky 2008].

- Model the probability of an event given prior events.

Background: Statistical Script Learning



Background: Statistical Script Inference



Outline

- Background
  - Statistical Scripts
  - Recurrent Neural Nets

Background: RNNs

- **Recurrent Neural Nets (RNNs)**: Neural Nets with cycles in computation graph.

- RNN Sequence Models: Map inputs

  $x_1, \ldots, x_t$

  to outputs

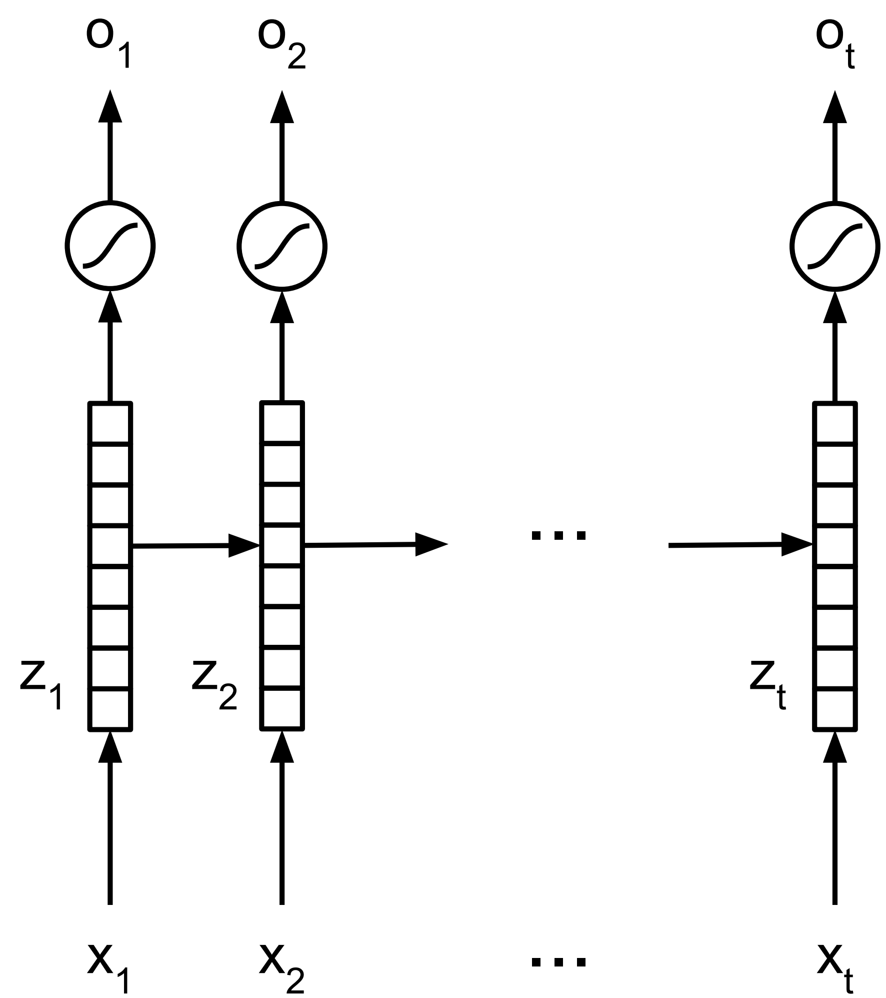
  $o_1, \ldots, o_t$

  via learned latent vector states

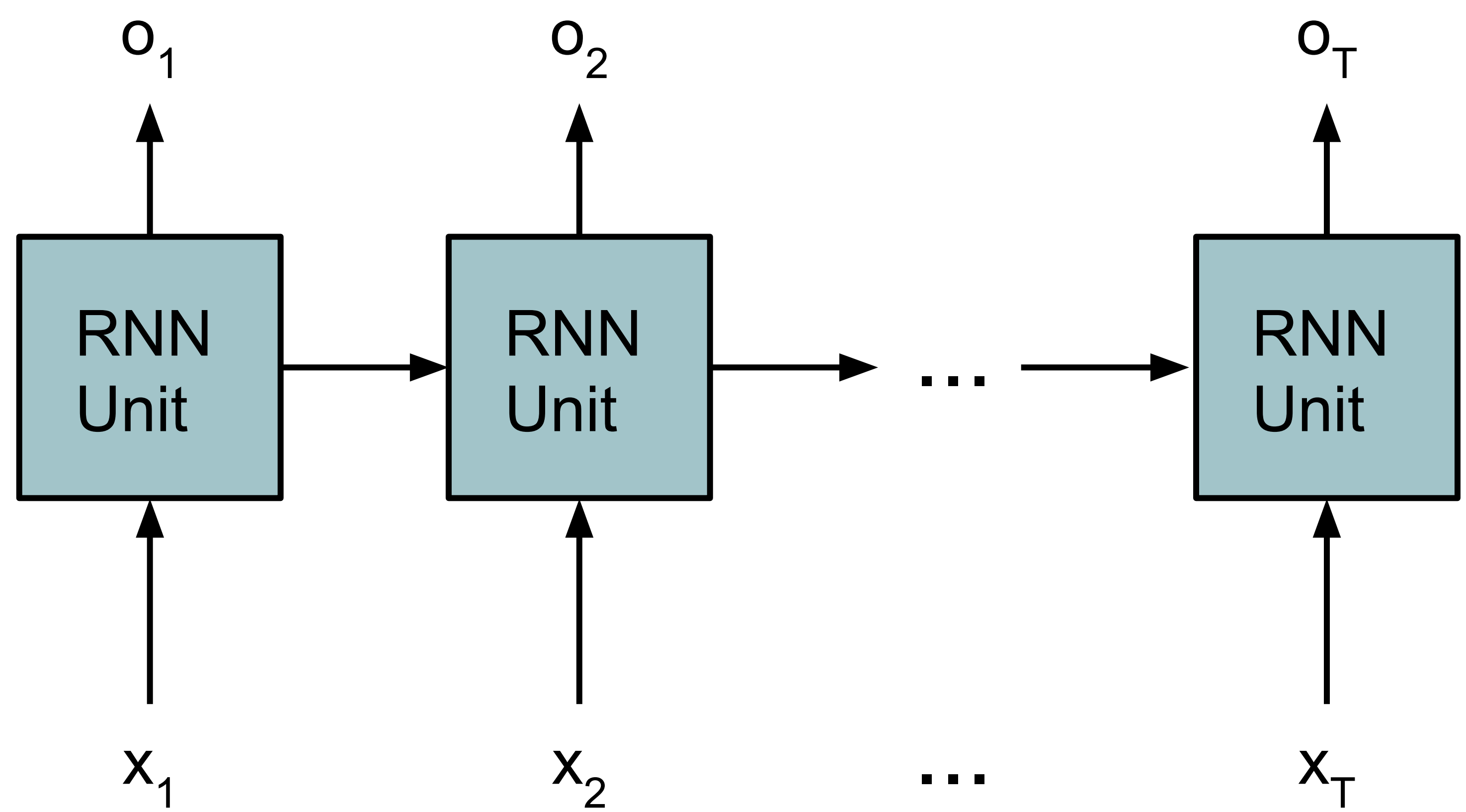
  $z_1, \ldots, z_t$.

Background: RNNs

[Elman 1990]



Background: RNNs



- Hidden Unit can be arbitrarily complicated, as long as we can calculate gradients!

Background: LSTMs

- **Long Short-Term Memory (LSTM):** More complex hidden RNN unit. [Hochreiter & Schmidhuber, 1997]
- Explicitly addresses two issues:
  - Vanishing Gradient Problem.
  - Long-Range Dependencies.

Background: LSTMs

- LSTMs recently successful on many hard NLP tasks:
  - Machine Translation [Kalchbrenner & Blunsom 2013, Bahdanau et al. 2015].
  - Captioning Images/Videos [Donahue et al. 2015, Venugopalan et al. 2015].
  - Language Modeling [Sundermeyer et al. 2012, Kim et al. 2016].
  - Question Answering [Hermann et al. 2015, Gao et al. 2015].
  - Parsing [Vinyals et al. 2015].

Outline

- Background
- Methods
- Experiments
- Conclusion

Outline

- Background
- Methods
- Experiments
- Conclusion

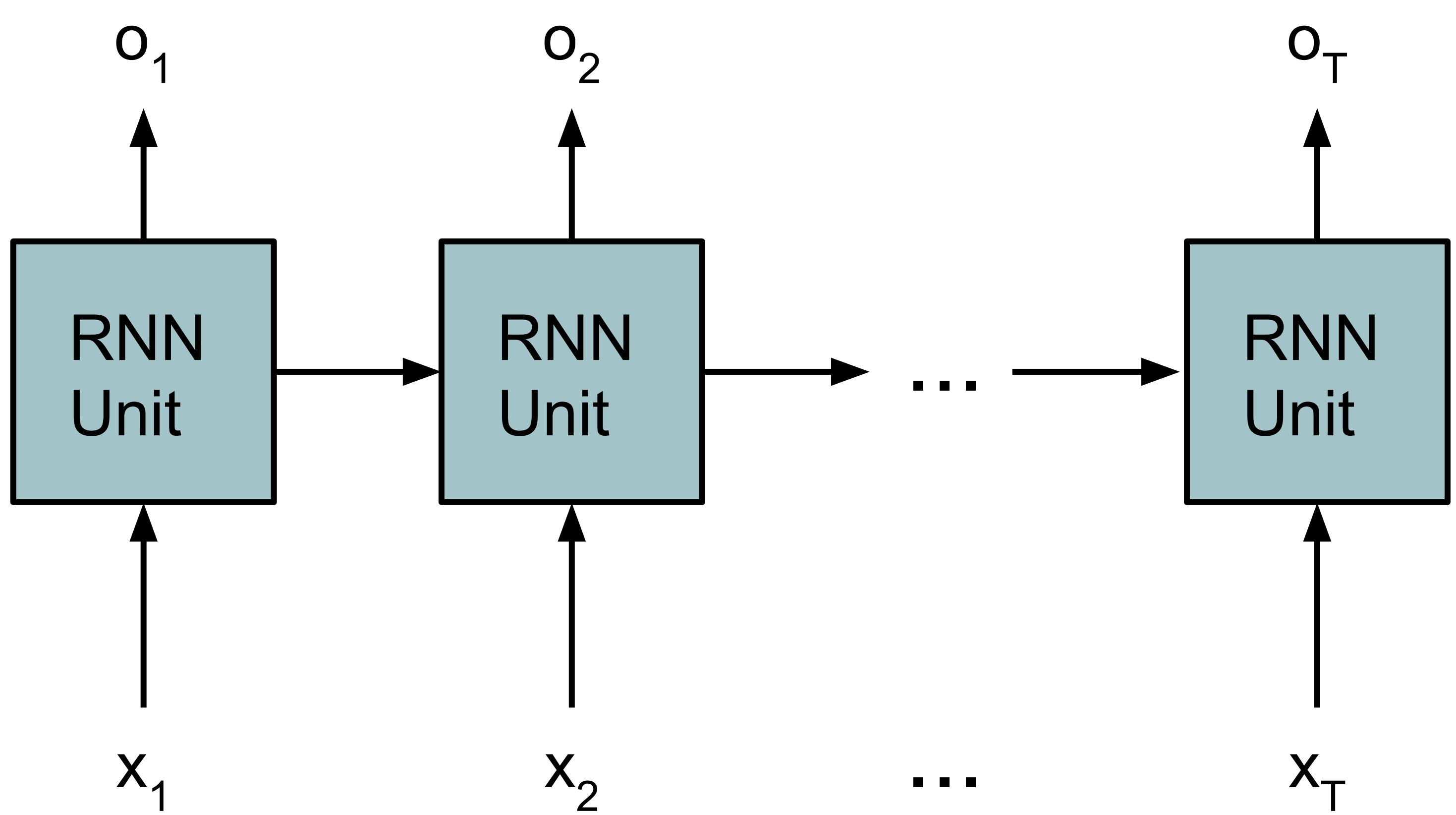
LSTM Script models

- Train LSTM sequence model on event sequences.
  - Events are (verbs + arguments).
  - Arguments can have noun info, coref info, or both.
- To infer events, the model generates likely events from sequence.

LSTM Script models

- *Mary's late husband Matthew, whom she married at 21 because she loved him, …*

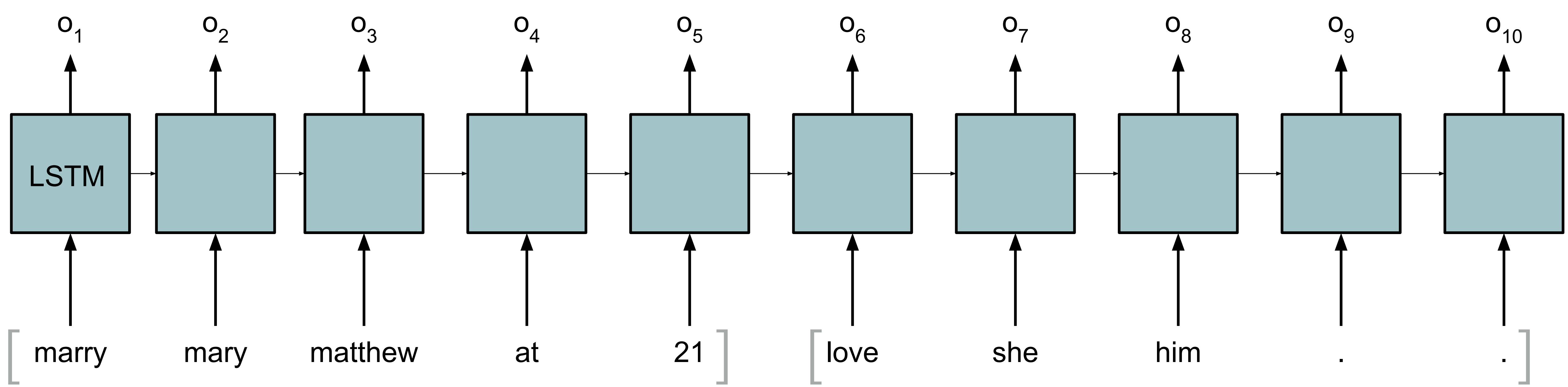
[marry, mary, matthew, at, 21]; [love, she, him]



LSTM Script models

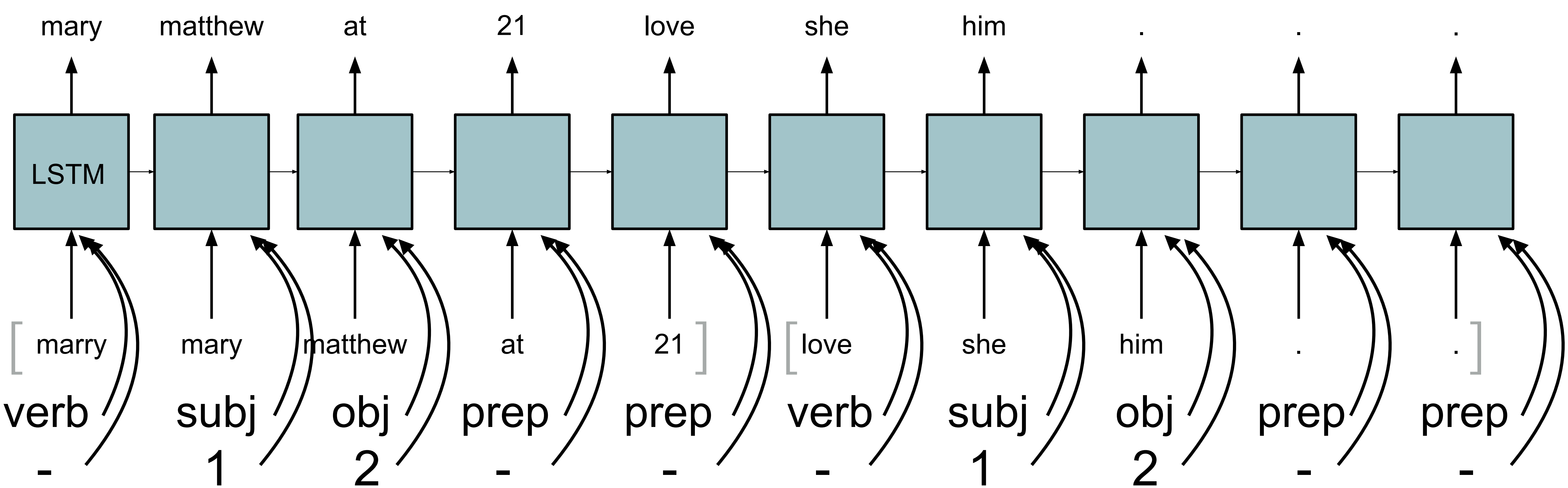
- *Mary's late husband Matthew, whom she married at 21 because she loved him, …*

[marry, mary, matthew, at, 21]; [love, she, him]



LSTM Script models

- *Mary's late husband Matthew, whom she married at 21 because she loved him, …*



Outline

- Background
- Methods
- Experiments
- Conclusion

Experimental Setup

- Train on English Wikipedia.
- Use Stanford CoreNLP to extract event sequences.
- Train LSTM using Batch Stochastic Gradient Descent with Momentum.
- To infer next events, have the LSTM generate additional events with highest probability.

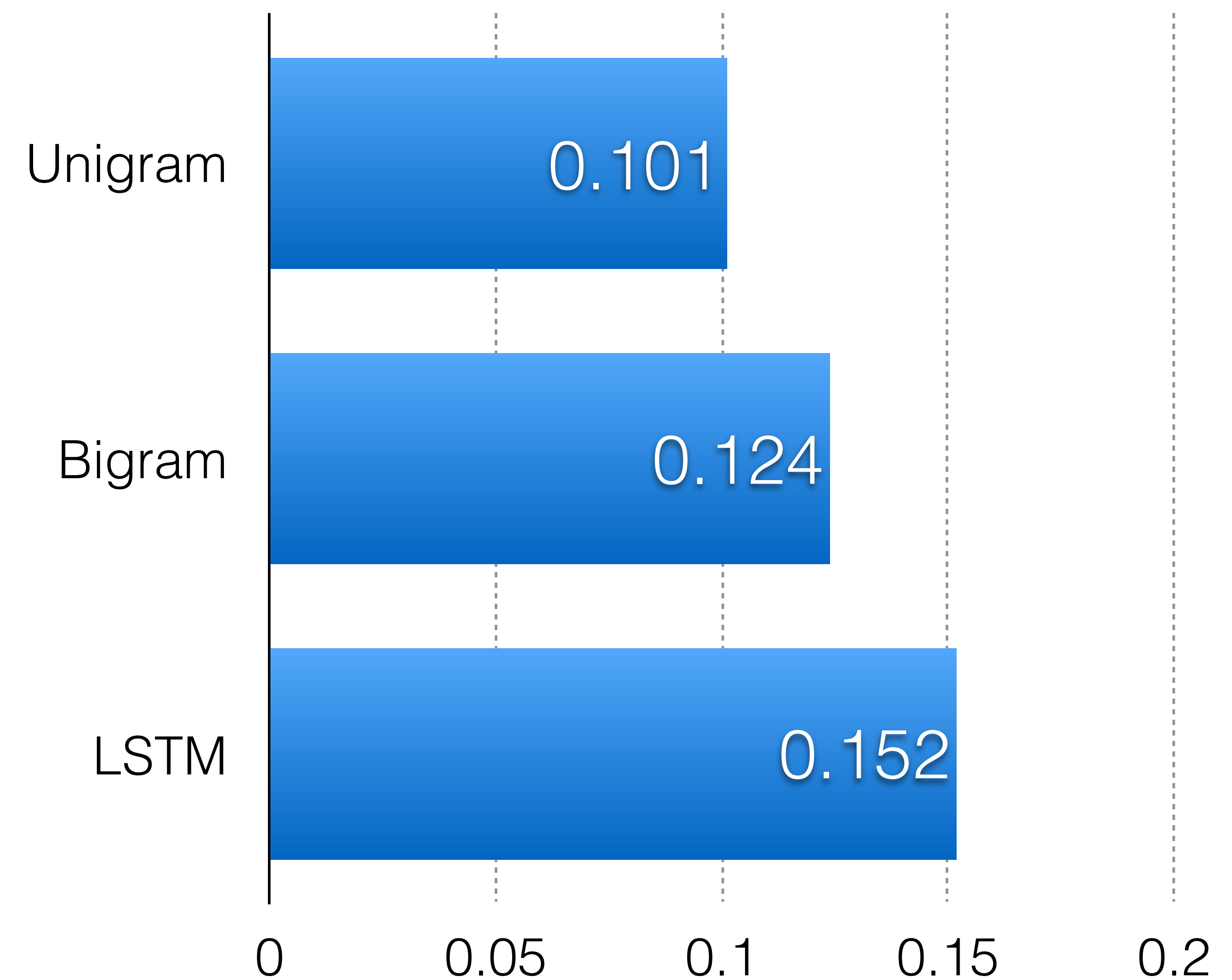
Evaluation

- "Narrative Cloze" (Chambers & Jurafsky, 2008): from an unseen document, hold one event out, try to infer it given remaining document.
- "Recall at *k*" (Jans et al., 2012): make *k* top inferences, calculate recall of held-out events.
- (More metrics in the paper.)

Evaluation

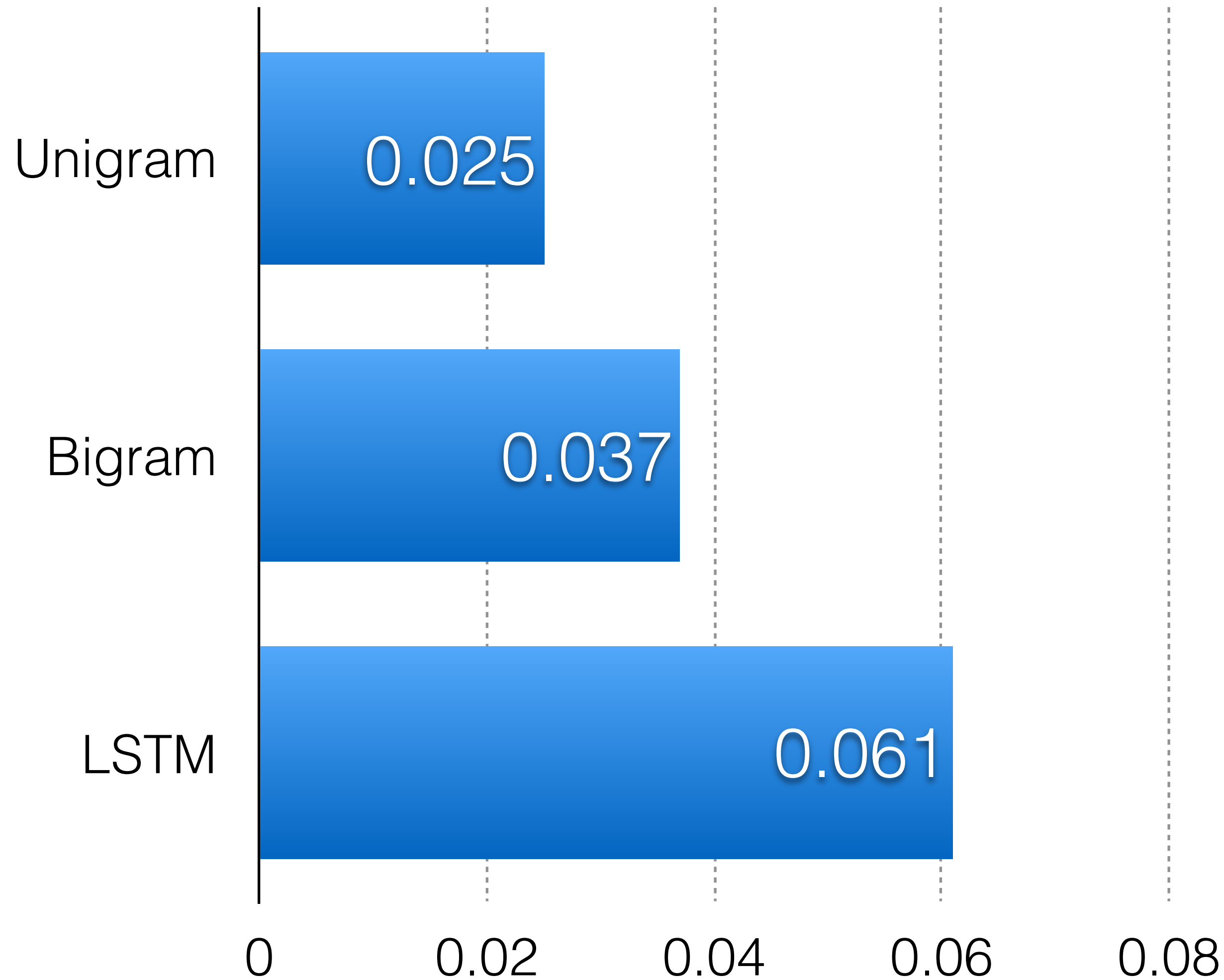
- Three Systems:
  - **Unigram:** Always guess most common events.
  - **Bigram:** Variations of Pichotta & Mooney (2014)
    - Uses event co-occurrence counts.
    - Best-published system on task.
  - **LSTM:** LSTM script system (this work).

Results: Predicting Verbs & Coreference Info



Recall at 25 for inferring Verbs & Coref info

Results: Predicting Verbs & Nouns

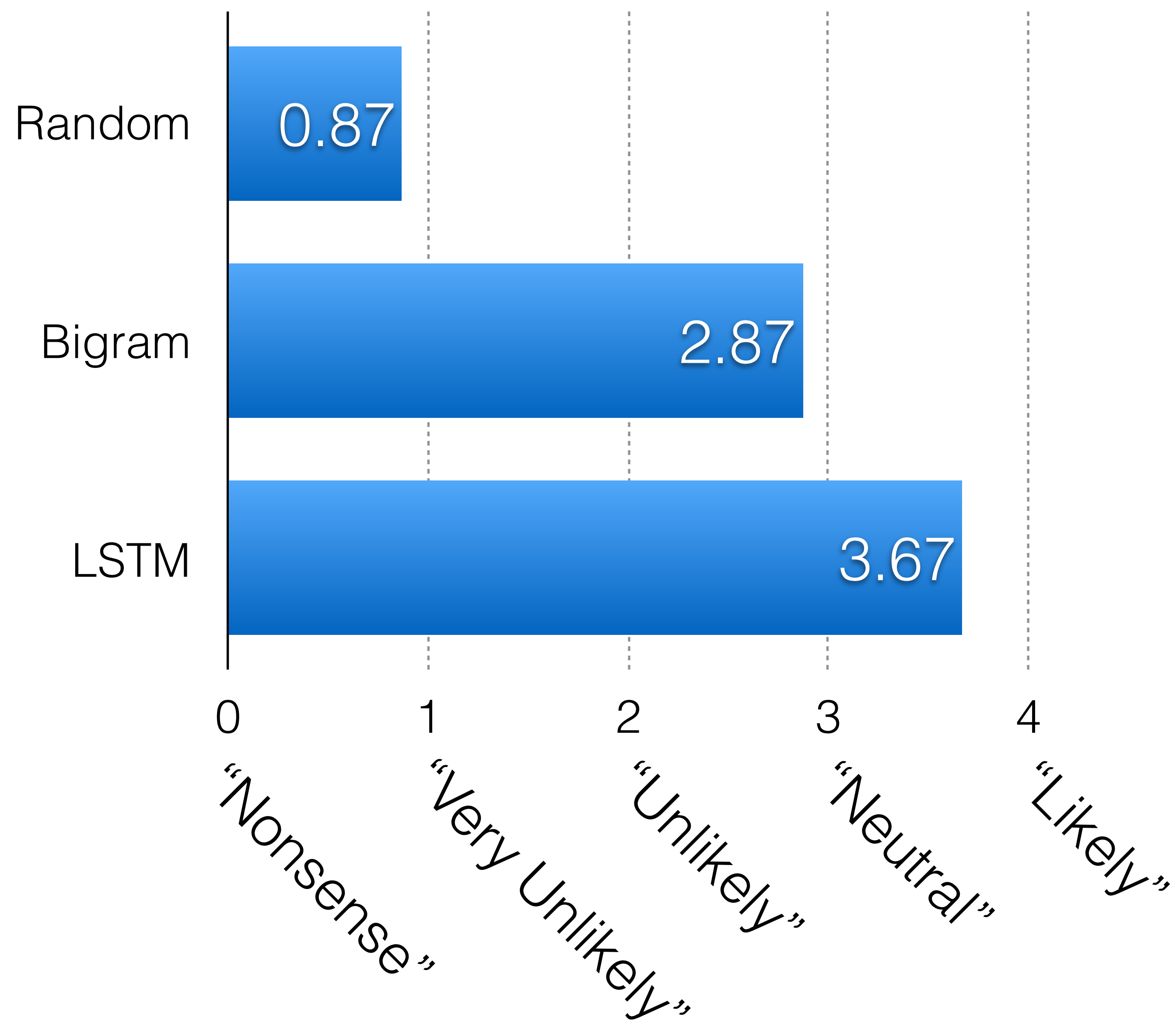


Recall at 25 for inferring Verbs & Nouns

Human Evaluations

- Solicit judgments on individual inferences on Amazon Mechanical Turk.
  - Have annotators rate inferences from 1-5 (or mark “Nonsense,” scored 0).
  - More interpretable.

Results: Crowdsourced Eval



Generated “Story”

Generated event tuples	English Descriptions
(bear, ., ., kingdom, into)	Born into a kingdom,…
(attend, she, brown, graduation, after)	…she attended Brown after graduation
(earn, she, master, university, from)	She earned her Masters from the University
(admit, ., she, university, to)	She was admitted to a University
(receive,she,bachelor,university,from)	She had received a bachelors from a University
(involve, ., she, production, in)	She was involved in the production
(represent, she, company, ., .)	She represented the company.

Conclusion

- Presented a method for inferring implicit events with LSTMs.
- Superior performance on reconstructing held-out events and inferring novel events.

Thanks!