Architectures and representations for string transduction

Grzegorz Chrupała • Tilburg University

CLIN 2015

Work-in-progress report and ideas
– not *results* talk

String transduction

procrastination proʊˌkræstɪˈneɪʃn	اصفهان Esfahan

i will c wat i can do
i will see what I can do

As sequence labeling

- Determine sequence of operations to convert input to output
- If input/output alphabets overlap – simple **diff** algorithm

Edit script

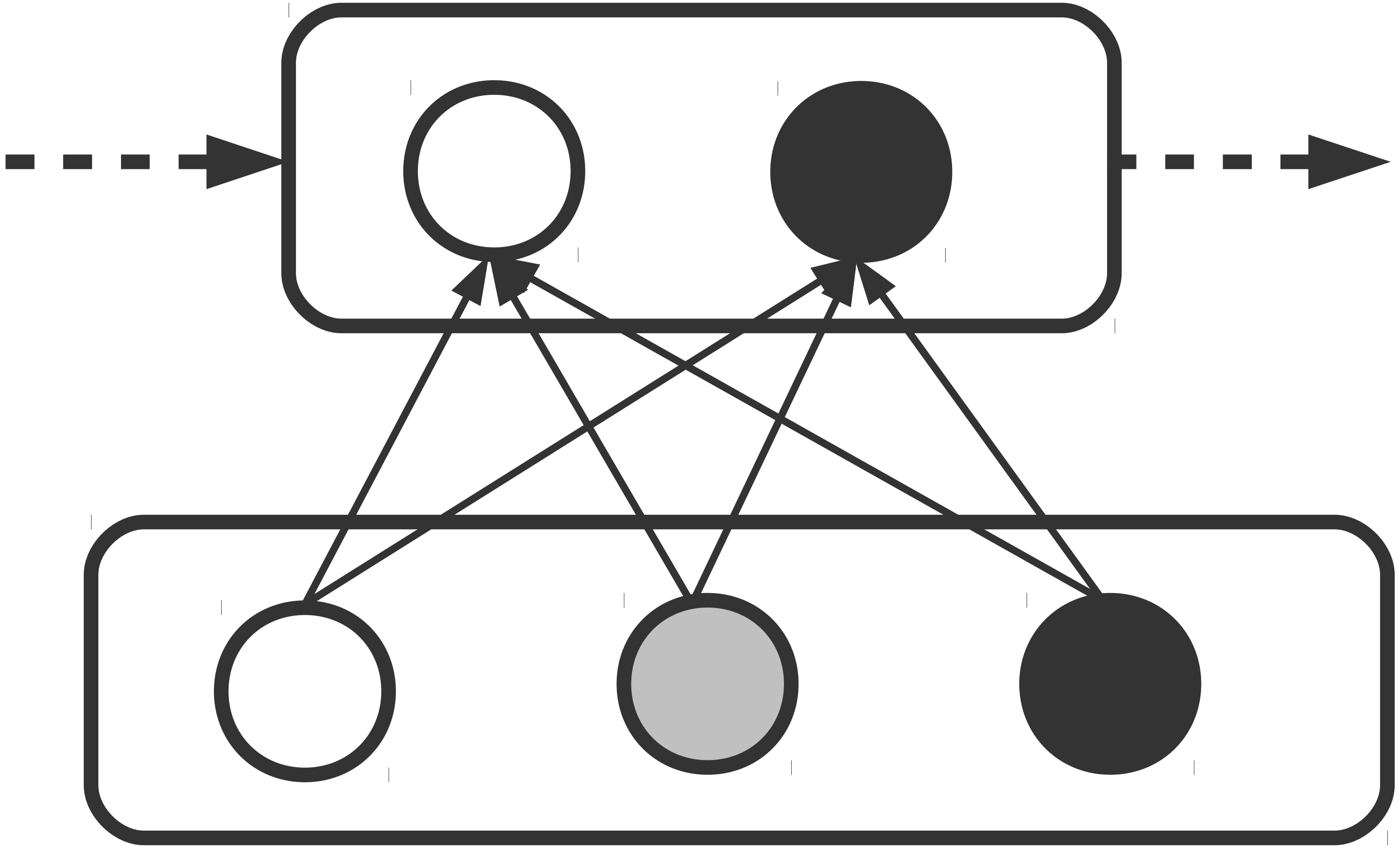
c	_	w	a	t
REP(c, see)	NIL	NIL	INS(h)	NIL
see	_	w	ha	t

Each position in the string labeled with an edit op

Sequence labelers

- Linear
  - **S**tructure **P**erceptron (Collins 2002)
  - **C**onditional **R**andom **F**ields (Lafferty et al 2001)
- Nonlinear (hidden layer)
  - **S**imple **R**ecurrent **N**etworks (Elman 1990)

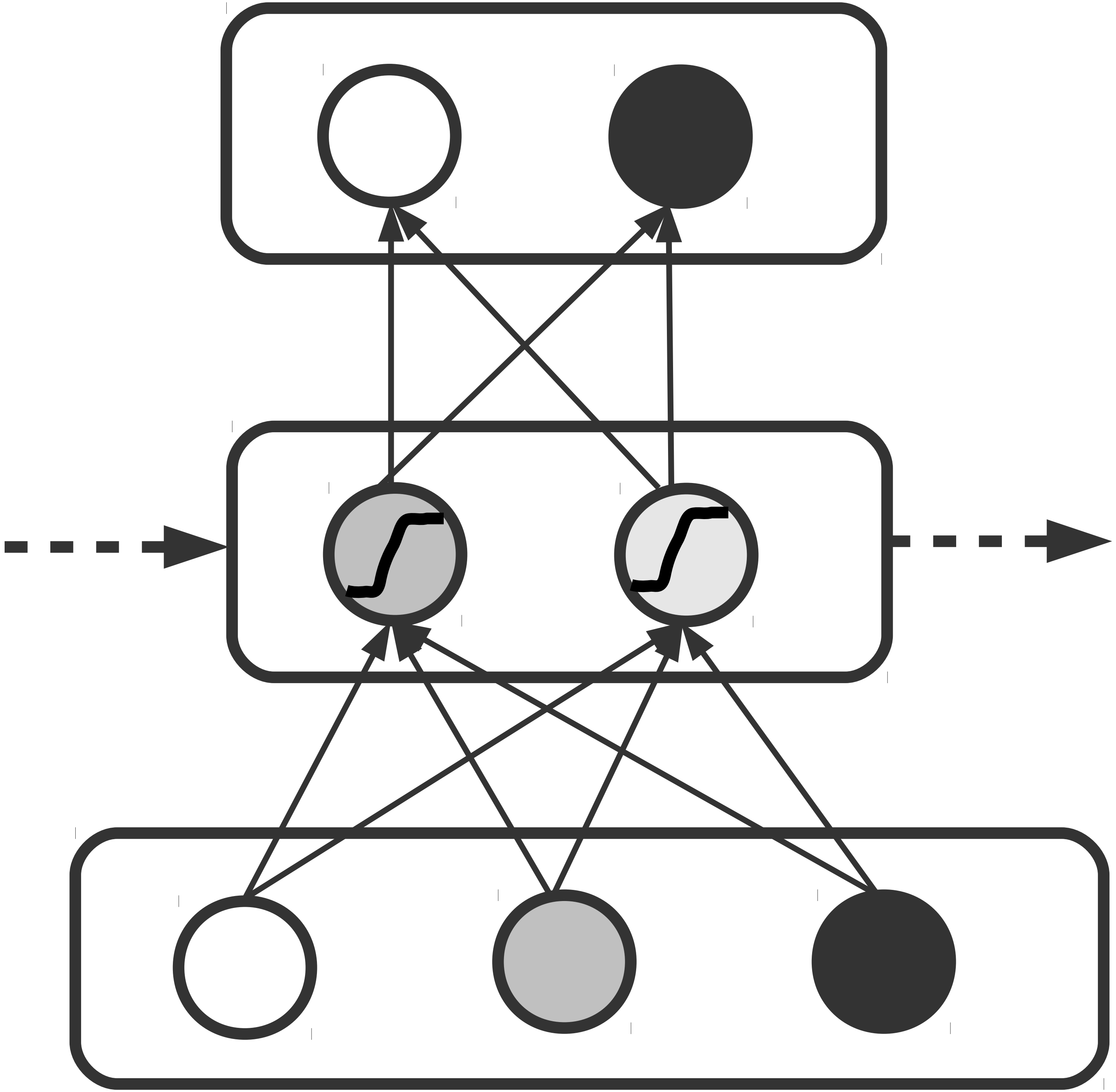
Linear models



Linear models – representations

- Need rich input
  - Large n-grams
  - (Word embeddings)
  - (Text embeddings)
  - (Dictionary features)

Non-linear models



Nonlinear models – representations

In theory, can learn own representations from raw input

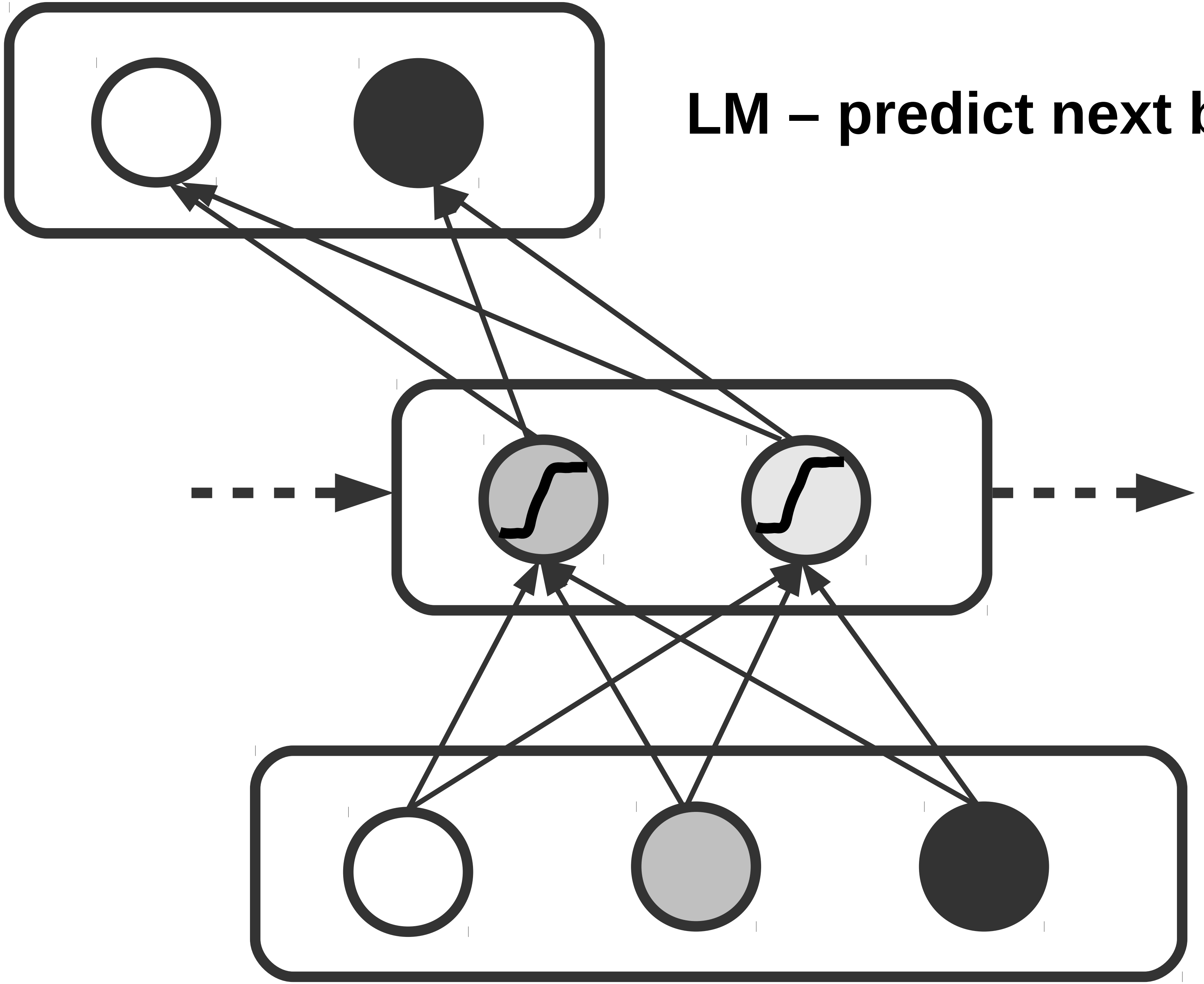
BUT well-known fact: **need lots of data**

Tweet normalization with few examples

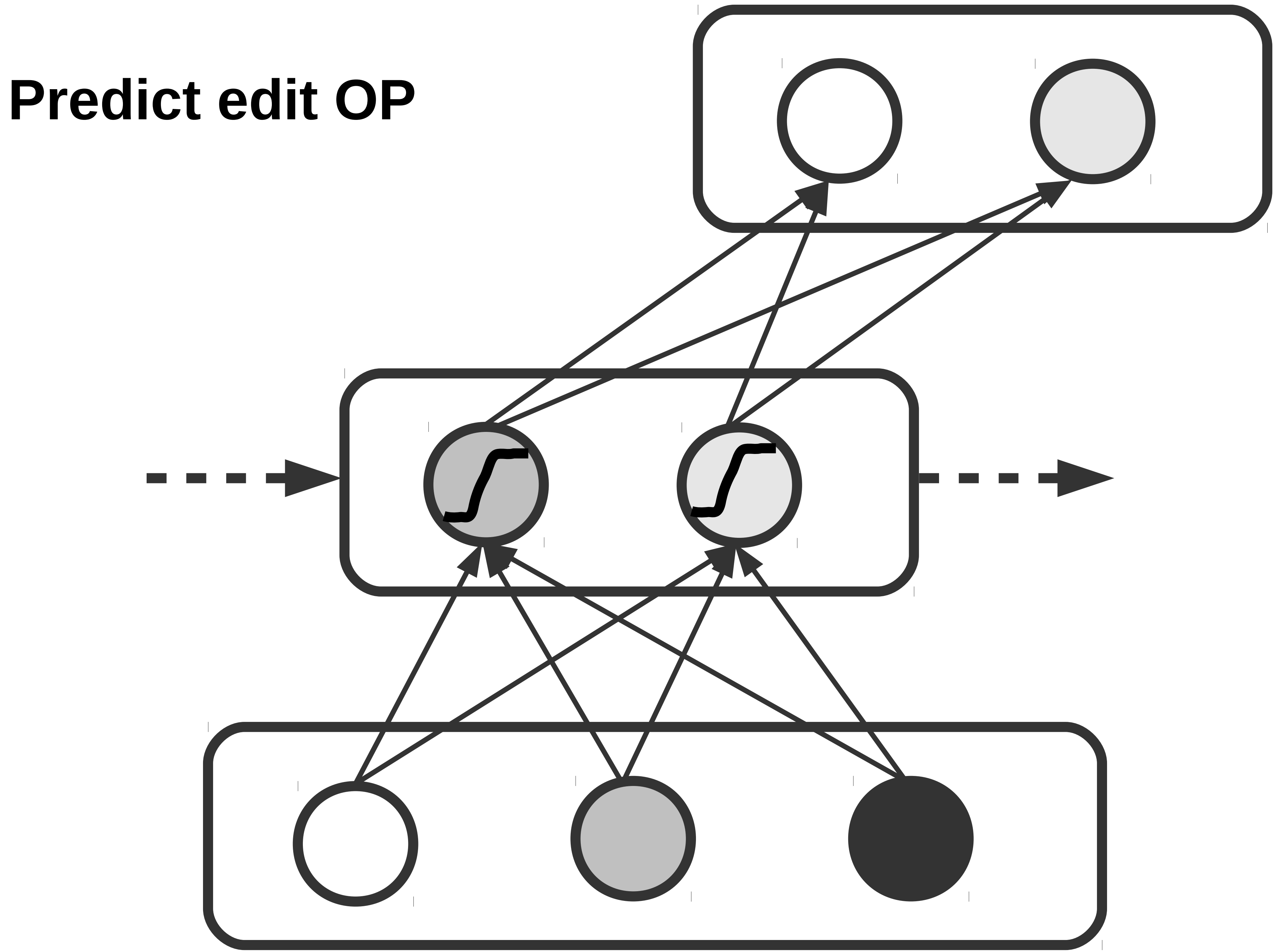
- Previously (ACL 2014)
  - Linear model for sequence labeling
  - (plus SRN to learn text embeddings)
- Today
  - Can SRN learn **anything** from only **500** tweets?

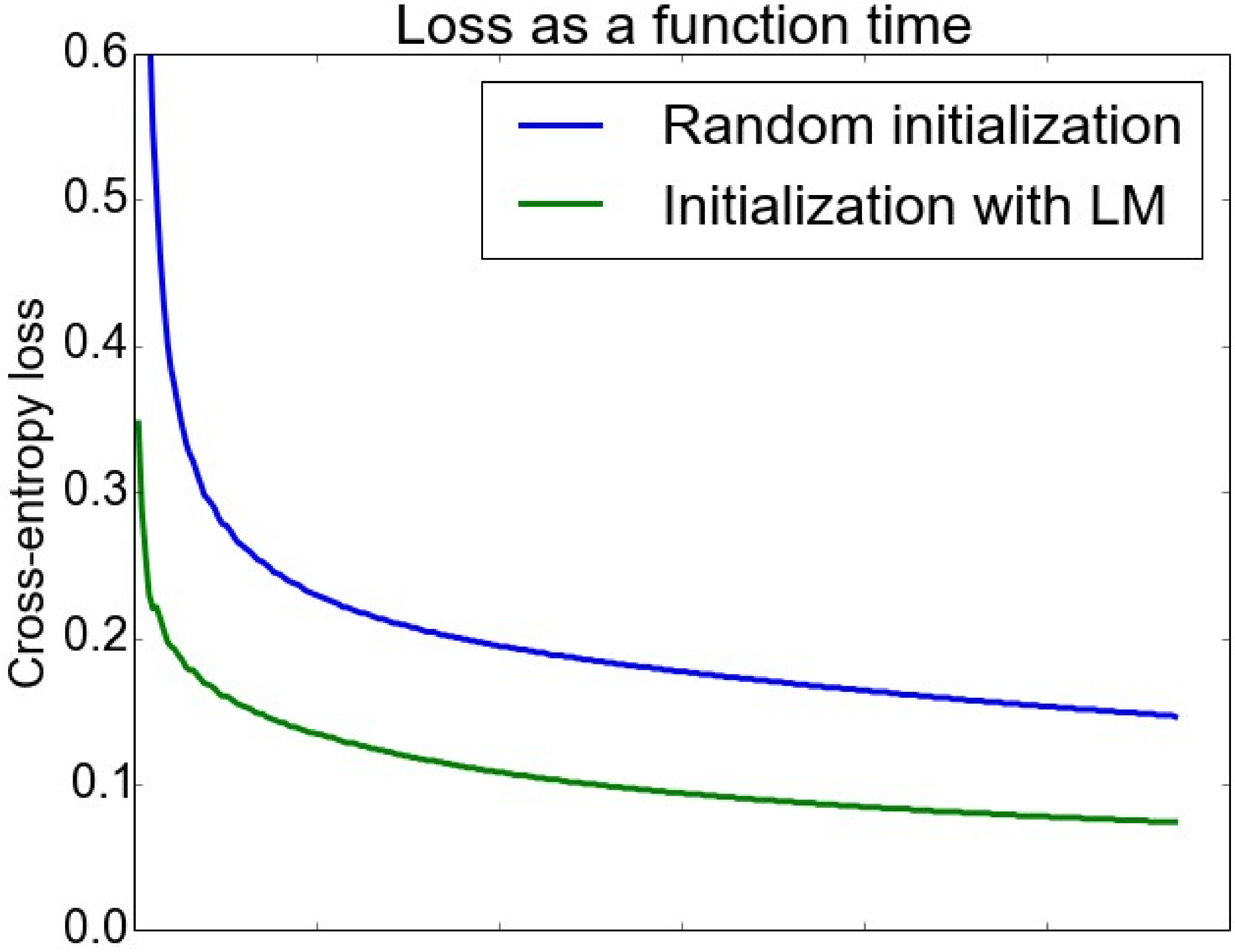
Pre-training

- Very simple form of multi-task learning
- Train a recurrent neural **L**anguage **M**odel on unlabeled text
- Use trained LM to **initialize** input-to-hidden weights

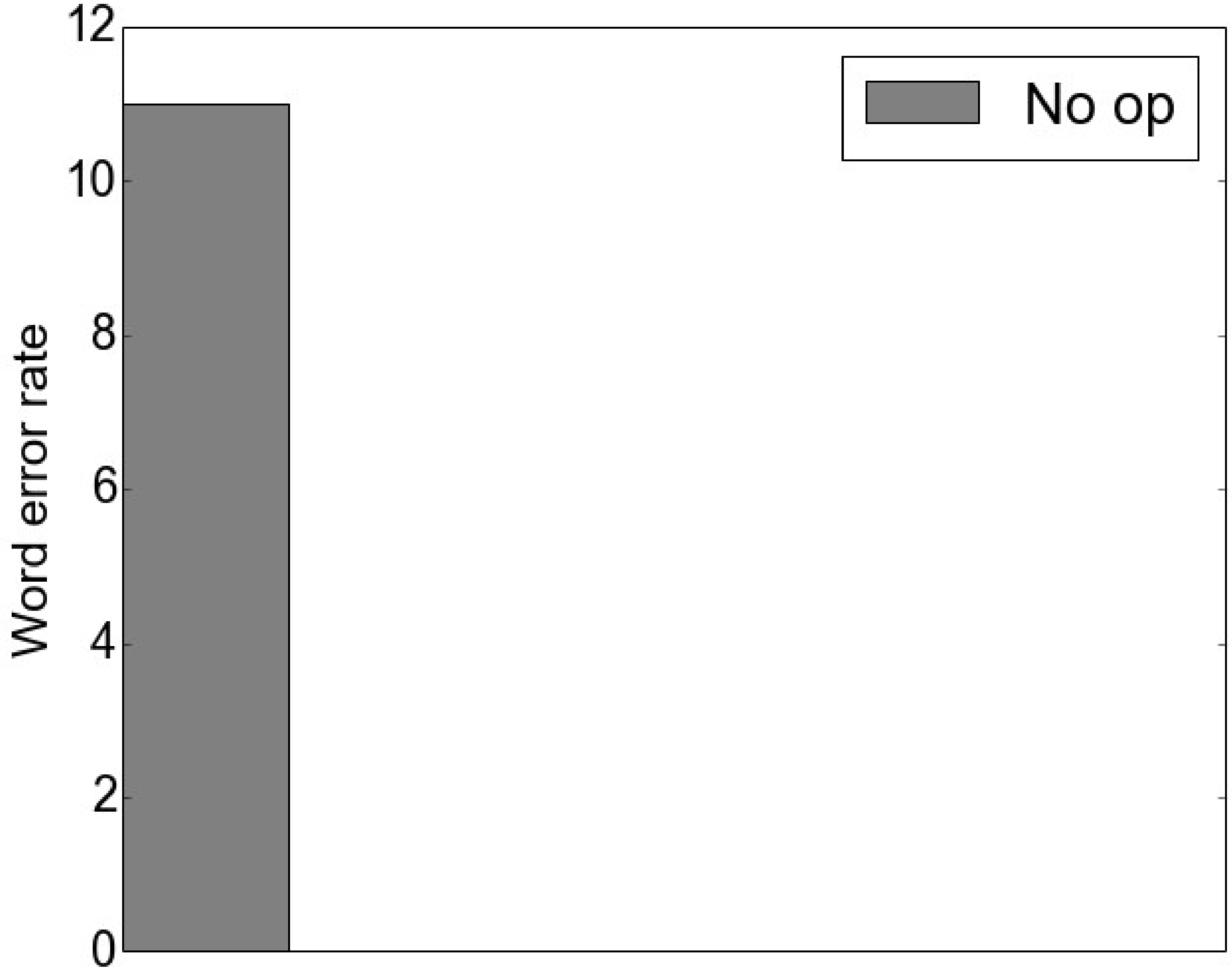


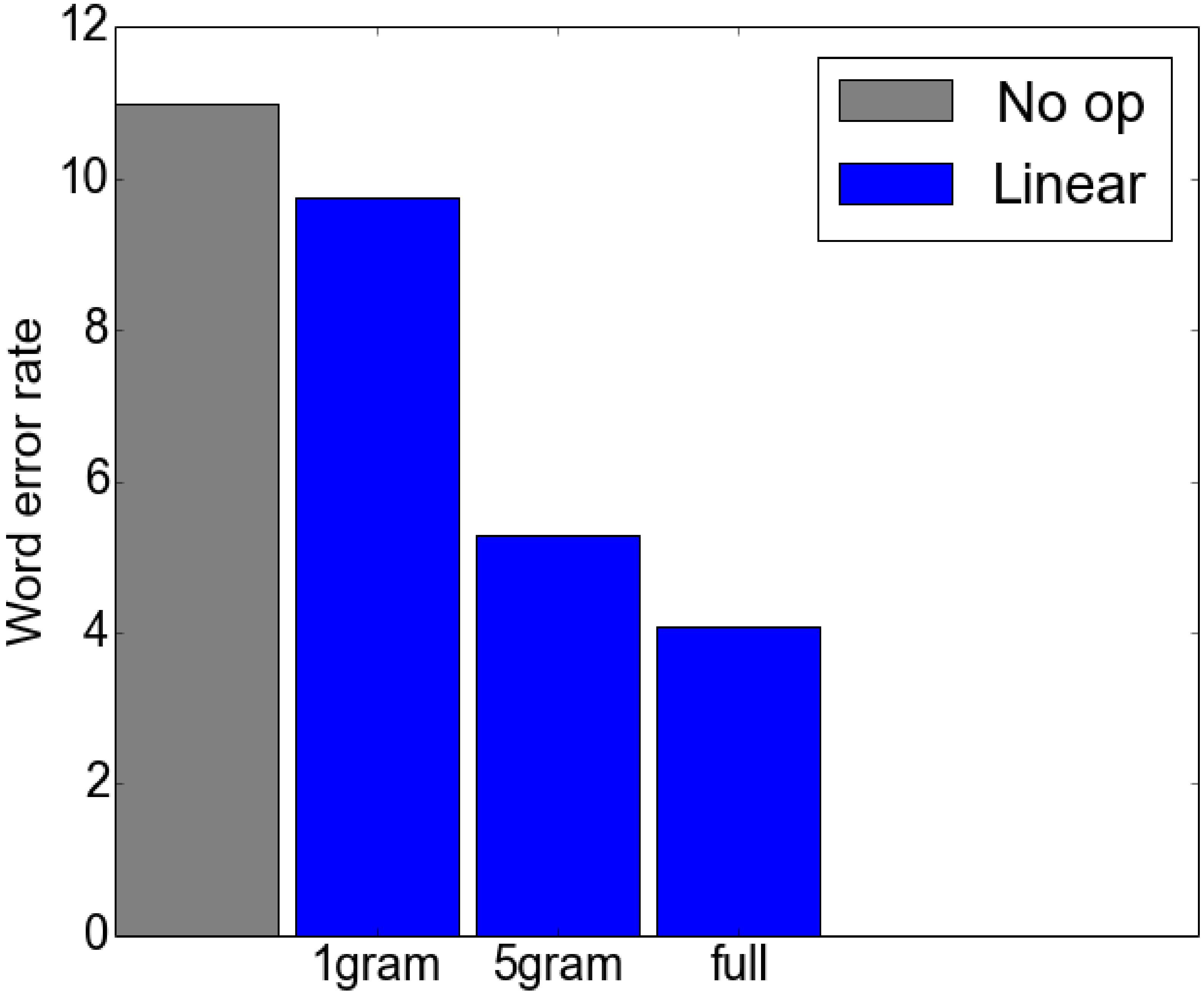
LM – predict next byte in sequence


Predict edit OP

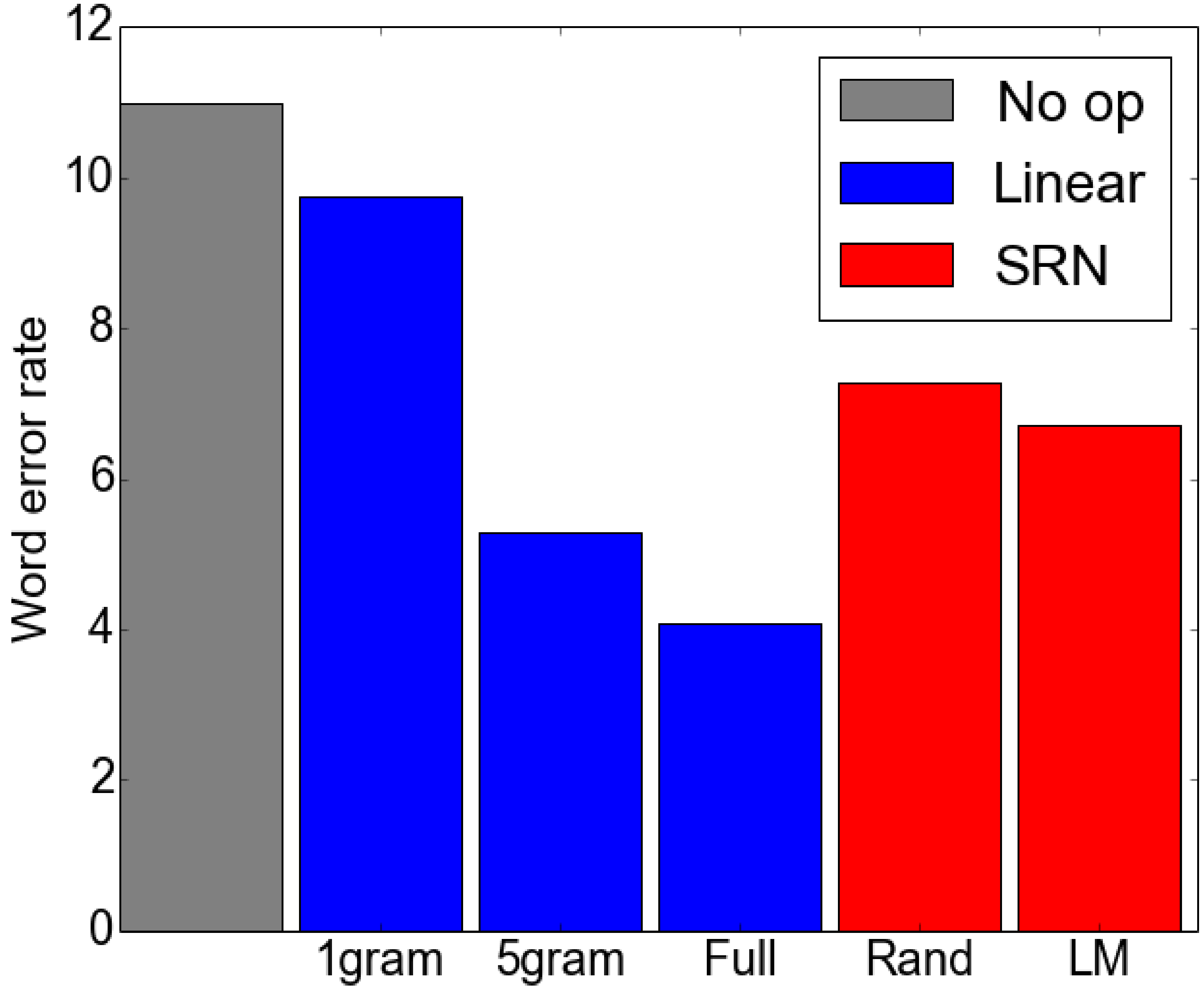


Loss as a function time
Cross-entropy loss
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Random initialization
Initialization with LM


12
10
8
6
4
2
0
Word error rate
No op



No op
Linear
Word error rate
0
2
4
6
8
10
12
1gram
5gram
full



Word error rate
0
2
4
6
8
10
12
1gram
5gram
Full
Rand
LM
No op
Linear
SRN

What next

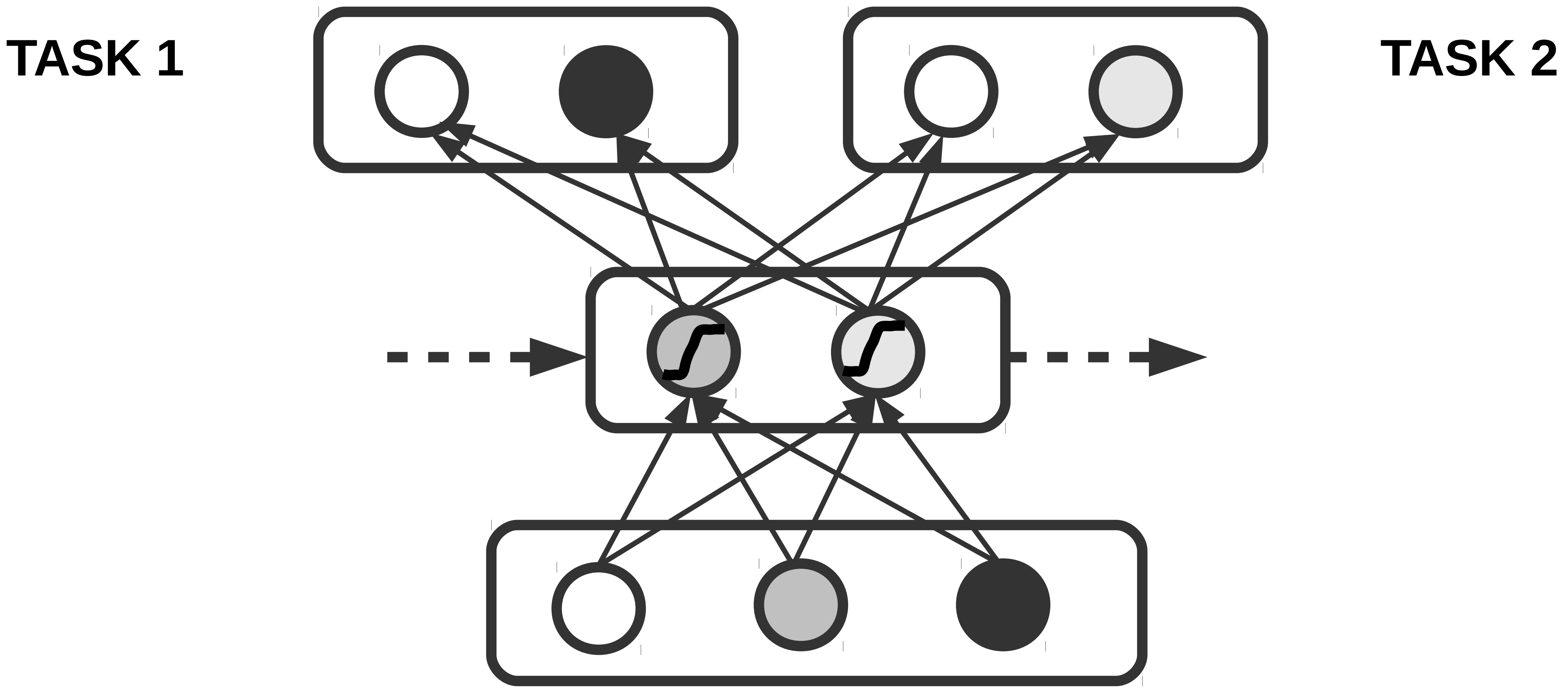
- Linear model works great
- Why bother with hidden layers et al?
  - Slow
  - Finicky learning rates, momentum, …

Large datasets

- Large datasets for normalization
  - Quite easy to annotate – 1000 text per day
  - **ACL 2015 Workshop on Noisy User-generated Text**
- Lots of data for grapheme to phoneme & transliteration

Reusable representations

Proper multi-task learning



Multi-task learning

(1) Language modeling
(2) Normalization
(3) Named entity recognition
(4) …

(2) and (3) at **Workshop Noisy User-generated Text**

Multimodal learning

Text + image with **shared representations**

(with Ákos Kádár and Afra Alishahi)

Conclusion

- Edit scripts + linear model + rich features
  - → effective text normalization
- Recurrent Network
  - benefits from LM
  - learns from very few examples
- Promising for large-scale multi-task learning