TTIC 31210:
Advanced Natural Language Processing

Kevin Gimpel

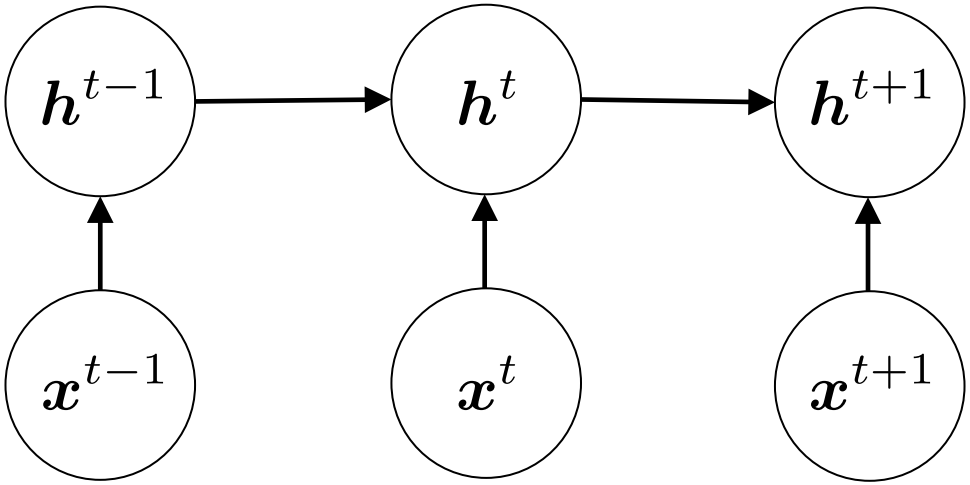
Spring 2017

Lecture 9:
NMT, finishing up Neural NLP

- Assignment 2 has been posted, due May 17
- Grades for Assignment 1 will be emailed to you soon
- Project proposal details posted, due May 10

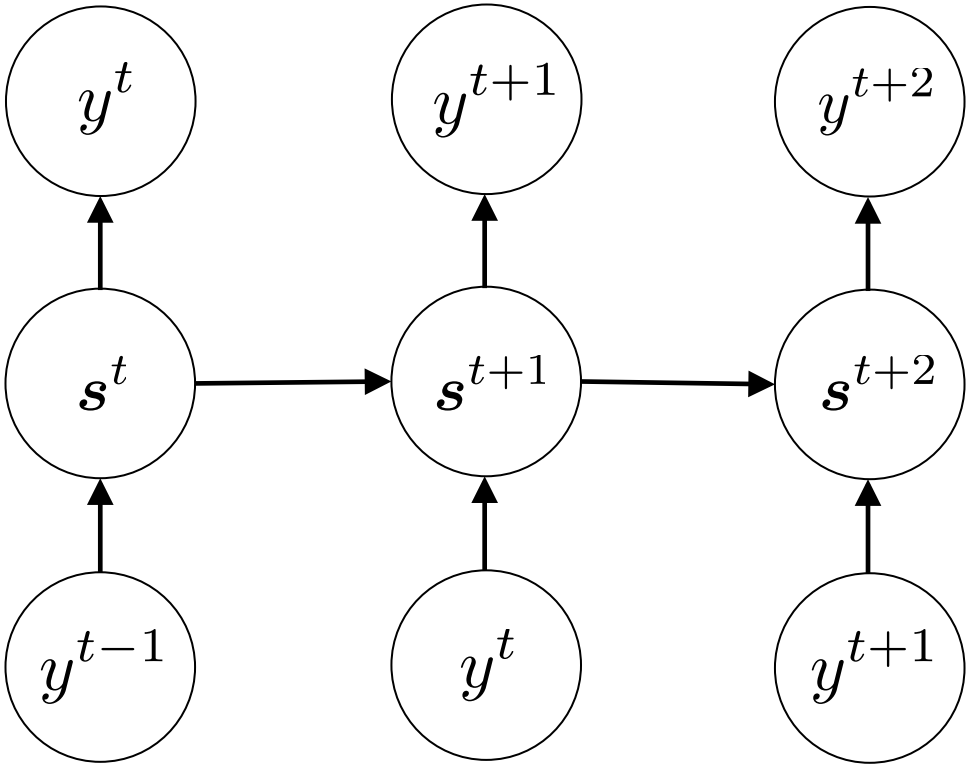
Input RNN (“Encoder”)

$$\boldsymbol{h}^t = \tanh\left(W^{(x)}\boldsymbol{x}^t + W^{(h)}\boldsymbol{h}^{t-1} + \boldsymbol{b}^{(h)}\right)$$



Output RNN ("Decoder")

$$y^t = \underset{y \in \mathcal{O}}{\operatorname{argmax}} \left(emb(y)^\top \boldsymbol{s}^t \right)$$



$$\boldsymbol{s}^t = \tanh \left(W^{(y)} \boldsymbol{y}^{t-1} + W^{(s)} \boldsymbol{s}^{t-1} + \boldsymbol{b}^{(s)} \right)$$

Learning

Minimum Risk Training

- "Bayes risk" is an alternative training objective that has been classically used in speech recognition and machine translation
- it permits the use of nearly-arbitrary evaluation metrics in training (through the specification of a "cost function")
- [on board]

"Minimum Risk Training for Neural Machine Translation"
Shen et al. (2016)

Minimum Risk Training

System	Training	MT06	MT02	MT03	MT04	MT05	MT08
MOSES	MERT	32.74	32.49	32.40	33.38	30.20	25.28
RNNSEARCH	MLE	30.70	35.13	33.73	34.58	31.76	23.57
	MRT	37.34	40.36	40.93	41.37	38.81	29.23

Table 3: Case-insensitive BLEU scores on Chinese-English translation.

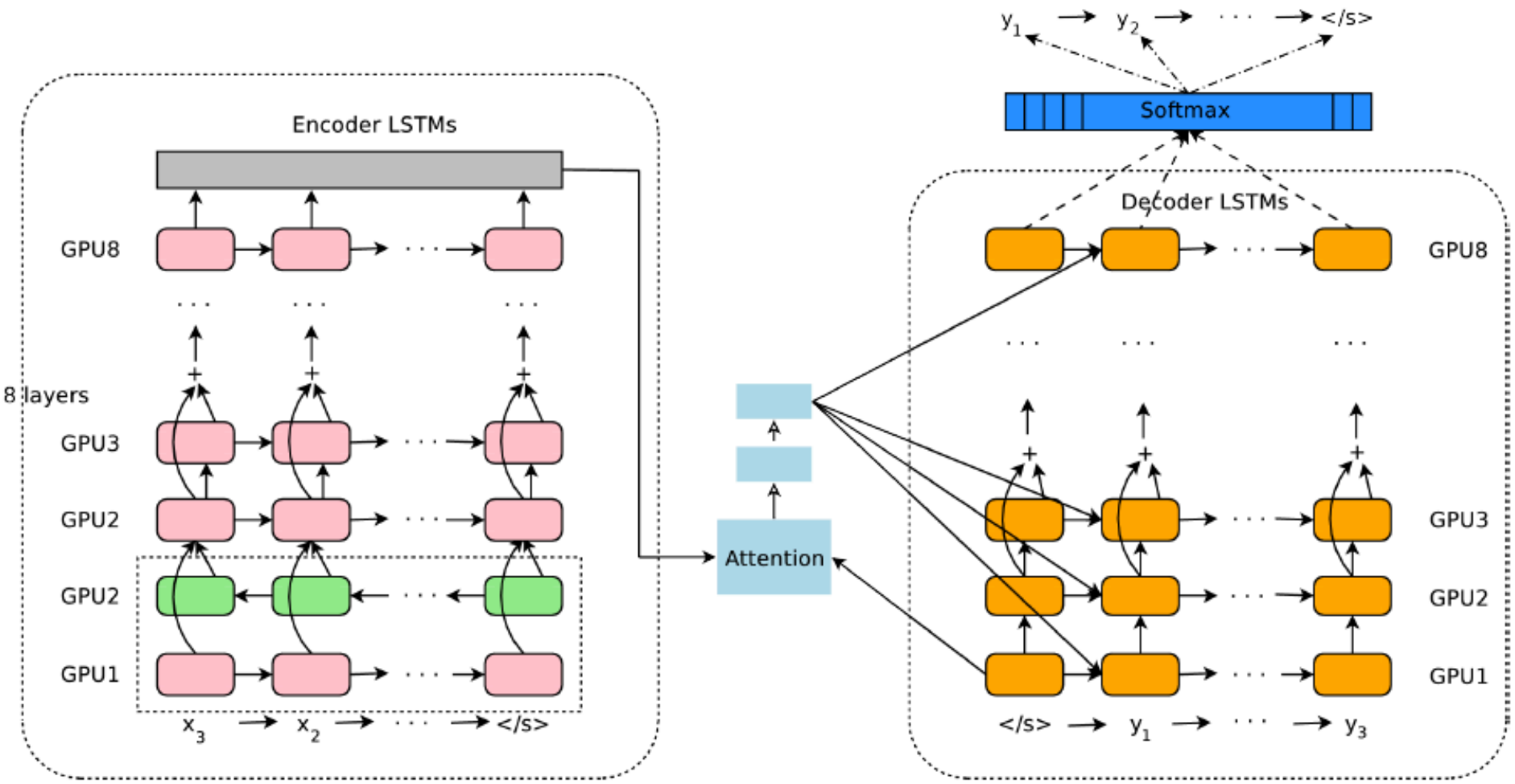
"Minimum Risk Training for Neural Machine Translation"
Shen et al. (2016)

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Google's NMT System



Google's NMT System

In our experience with large-scale translation tasks, simple stacked LSTM layers work well up to 4 layers, barely with 6 layers, and very poorly beyond 8 layers.

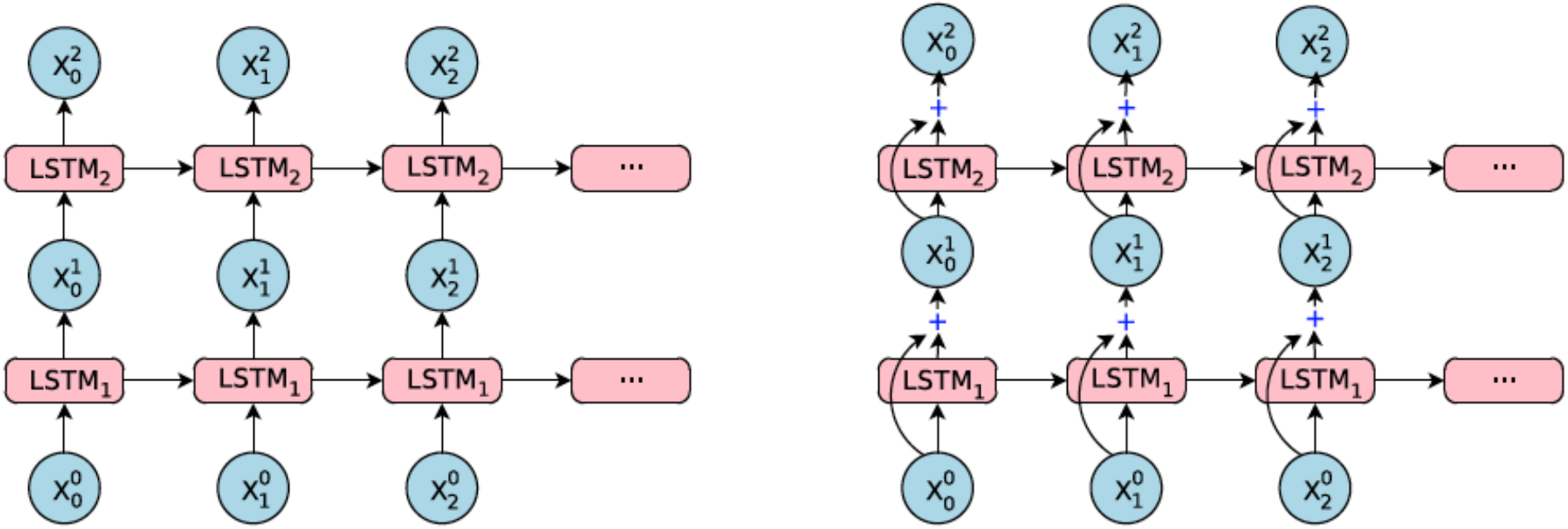


Figure 2: The difference between normal stacked LSTM and our stacked LSTM with residual connections. On the left: simple stacked LSTM layers [41]. On the right: our implementation of stacked LSTM layers with residual connections. With residual connections, input to the bottom LSTM layer ($\mathbf{x_i^0}$'s to LSTM$_1$) is element-wise added to the output from the bottom layer ($\mathbf{x_i^1}$'s). This sum is then fed to the top LSTM layer (LSTM$_2$) as the new input.

Google's NMT System

Here is an example of a word sequence and the corresponding wordpiece sequence:

- **Word**: Jet makers feud over seat width with big orders at stake
- **wordpieces**: _J et _makers _fe ud _over _seat _width _with _big _orders _at _stake

In the above example, the word "Jet" is broken into two wordpieces "_J" and "et", and the word "feud" is broken into two wordpieces "_fe" and "ud". The other words remain as single wordpieces. "_" is a special character added to mark the beginning of a word.

they use a procedure that deterministically segments any character sequence into wordpieces

vocab: 8k-32k wordpieces

they first learn a "wordpiece model"
we'll talk more about this sort of thing in the coming weeks

Google's NMT System

Table 5: Single model results on WMT En→De (newstest2014)

Model	BLEU	CPU decoding time per sentence (s)
Word	23.12	0.2972
Character (512 nodes)	22.62	0.8011
WPM-8K	23.50	0.2079
WPM-16K	24.36	0.1931
WPM-32K	24.61	0.1882
Mixed Word/Character	24.17	0.3268
PBMT [6]	20.7	
RNNSearch [37]	16.5	
RNNSearch-LV [37]	16.9	
RNNSearch-LV [37]	16.9	
Deep-Att [45]	20.6	

Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu,
Zhifeng Chen, Nikhil Thorat
melvinp,schuster,qvl,krikun,yonghui,zhifengc,nsthorat@google.com

Fernanda Viégas, Martin Wattenberg, Greg Corrado,
Macduff Hughes, Jeffrey Dean

To be able to make use of multilingual data within a single system, we propose one simple modification to the input data, which is to introduce an artificial token at the beginning of the input sentence to indicate the target language the model should translate to. For instance, consider the following English→Spanish pair of sentences:

```
Hello, how are you? -> ¿Hola como estás?
```

It will be modified to:

```
<2es> Hello, how are you? -> ¿Hola como estás?
```

Table 5: Portuguese→Spanish BLEU scores using various models.

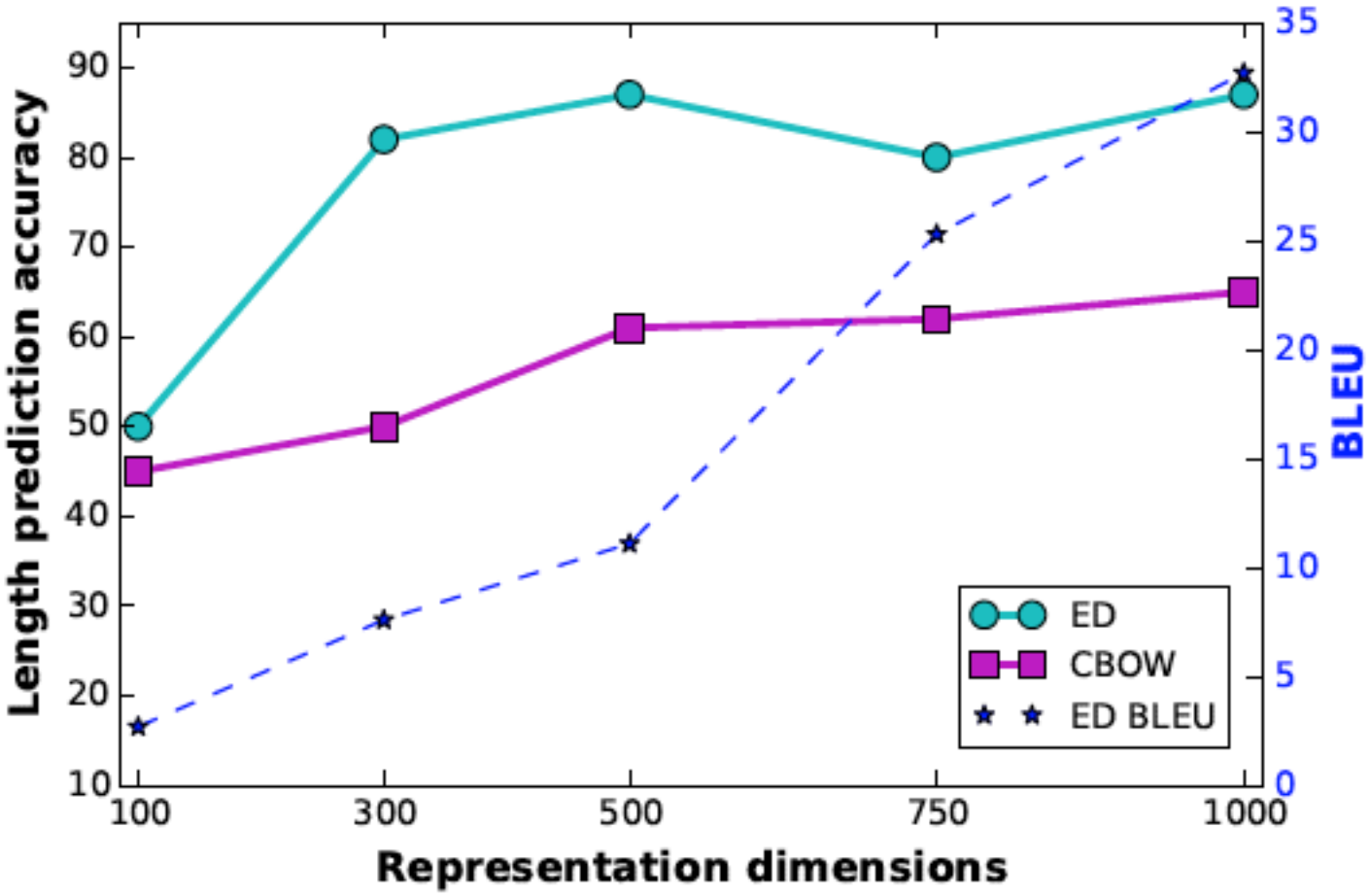
	Model	BLEU
(a)	PBMT bridged	28.99
(b)	NMT bridged	30.91
(c)	NMT Pt→Es	31.50
(d)	Model 1 (Pt→En, En→Es)	21.62
(e)	Model 2 (En↔{Es, Pt})	24.75
(f)	Model 2 + incremental training	31.77

Understanding Neural NLP

How do we understand what information is captured in an embedding?

- one approach is to attempt to predict different things and see what the accuracy is
- e.g., from a sentence embedding, attempt to predict sentence length, words in sentence, and word order in sentence (Adi et al., 2017)
- they compare word averaging ("CBOW") and an LSTM ("ED")

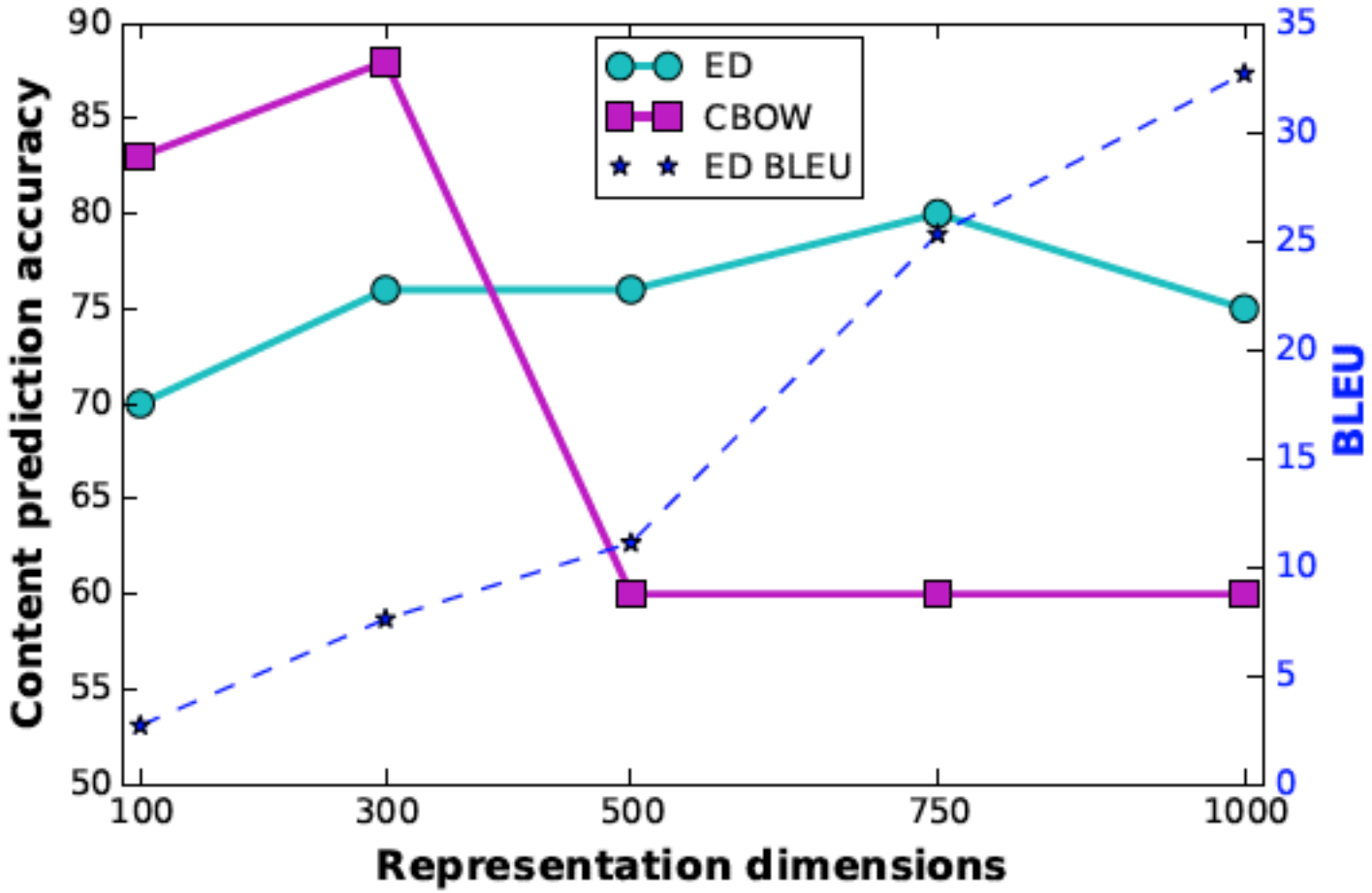
Predicting Sentence Length from Sentence Embeddings



length prediction = predict the length bin of the sentence, using 8 length bins (majority baseline gives 20% accuracy)

Adi et al. (ICLR 2017)

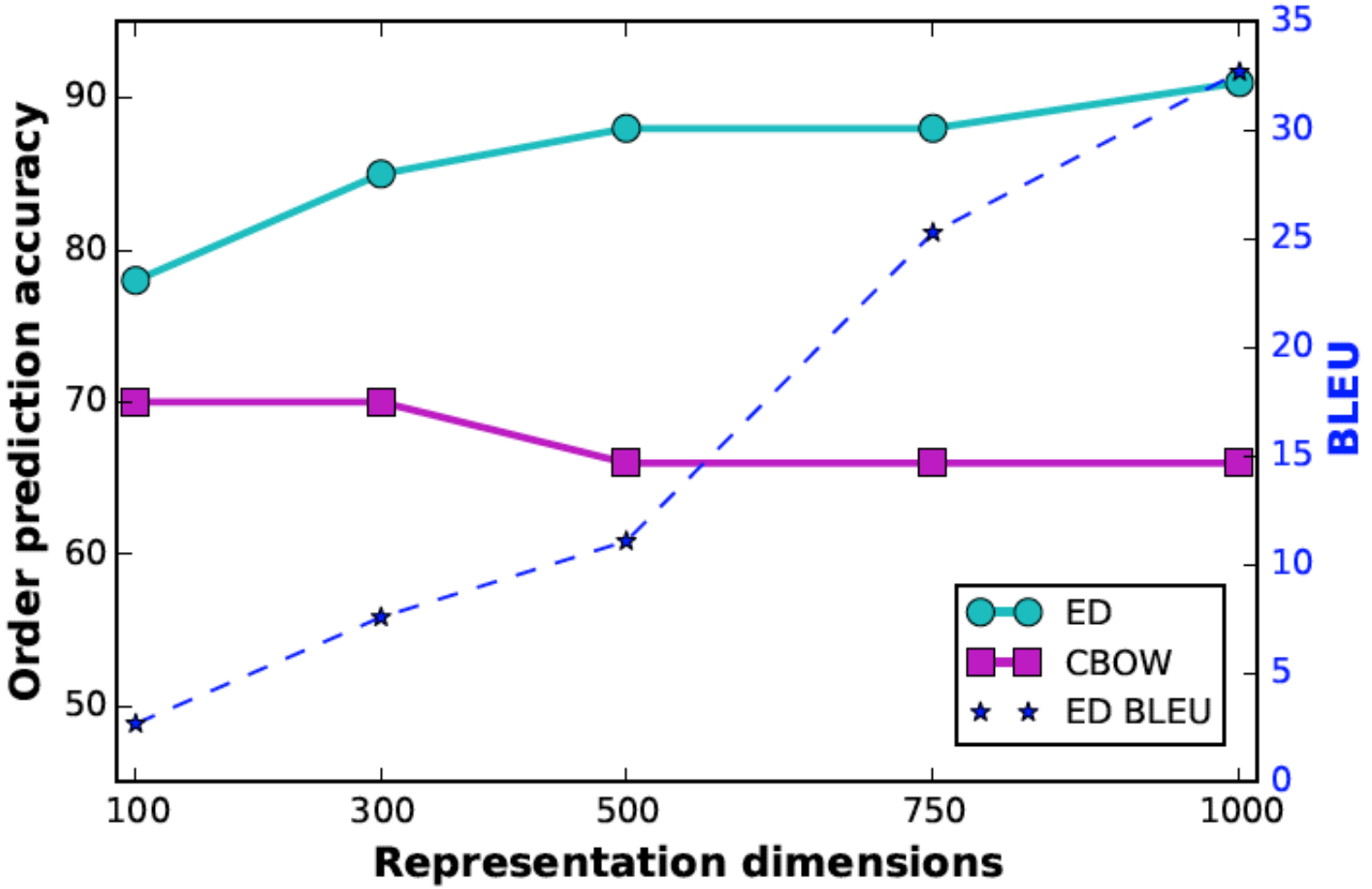
Predicting Words in Sentence ("Content") from Sentence Embeddings



content prediction = predict whether a word is contained in the sentence (baseline is 50%)

Adi et al. (ICLR 2017)

Predicting Word Order from Sentence Embeddings



order prediction = given two random words in sentence, predict which comes first (baseline is 50%)

Adi et al. (ICLR 2017)

Do LSTMs capture syntactic agreement?

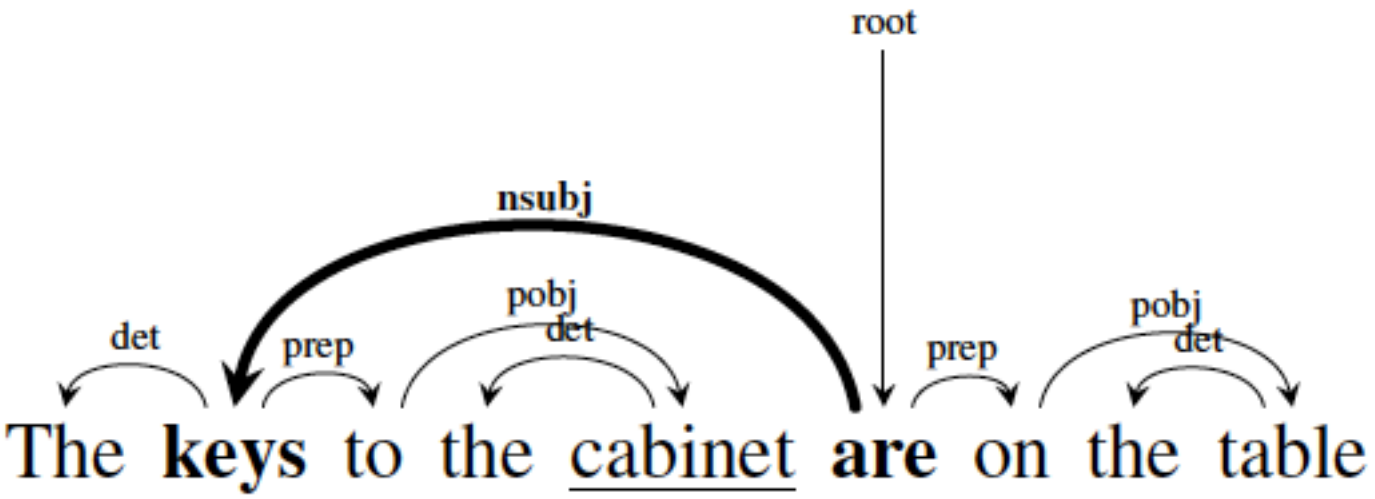
a. The **key** **is** on the table.
b. *The **key** **are** on the table.
c. *The **keys** **is** on the table.
d. The **keys** **are** on the table.

The **keys** to the cabinet **are** on the table.

Linzen et al. (TACL 2016)

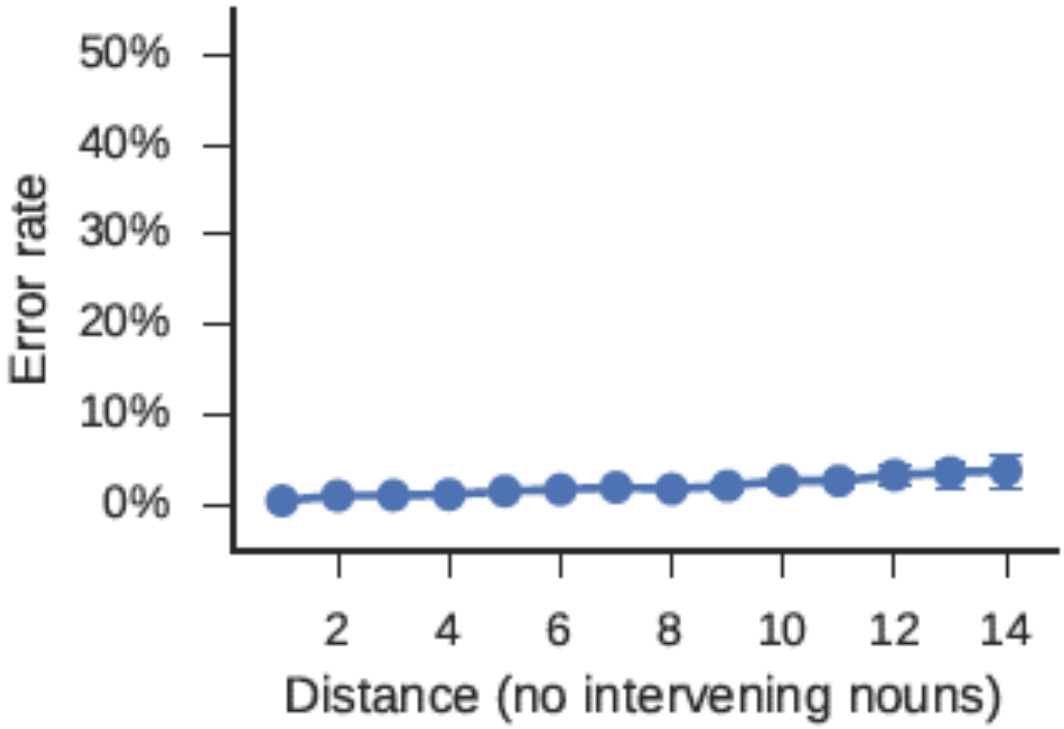
Do LSTMs capture syntactic agreement?

a. The **key is** on the table.
b. *The **key are** on the table.
c. *The **keys is** on the table.
d. The **keys are** on the table.



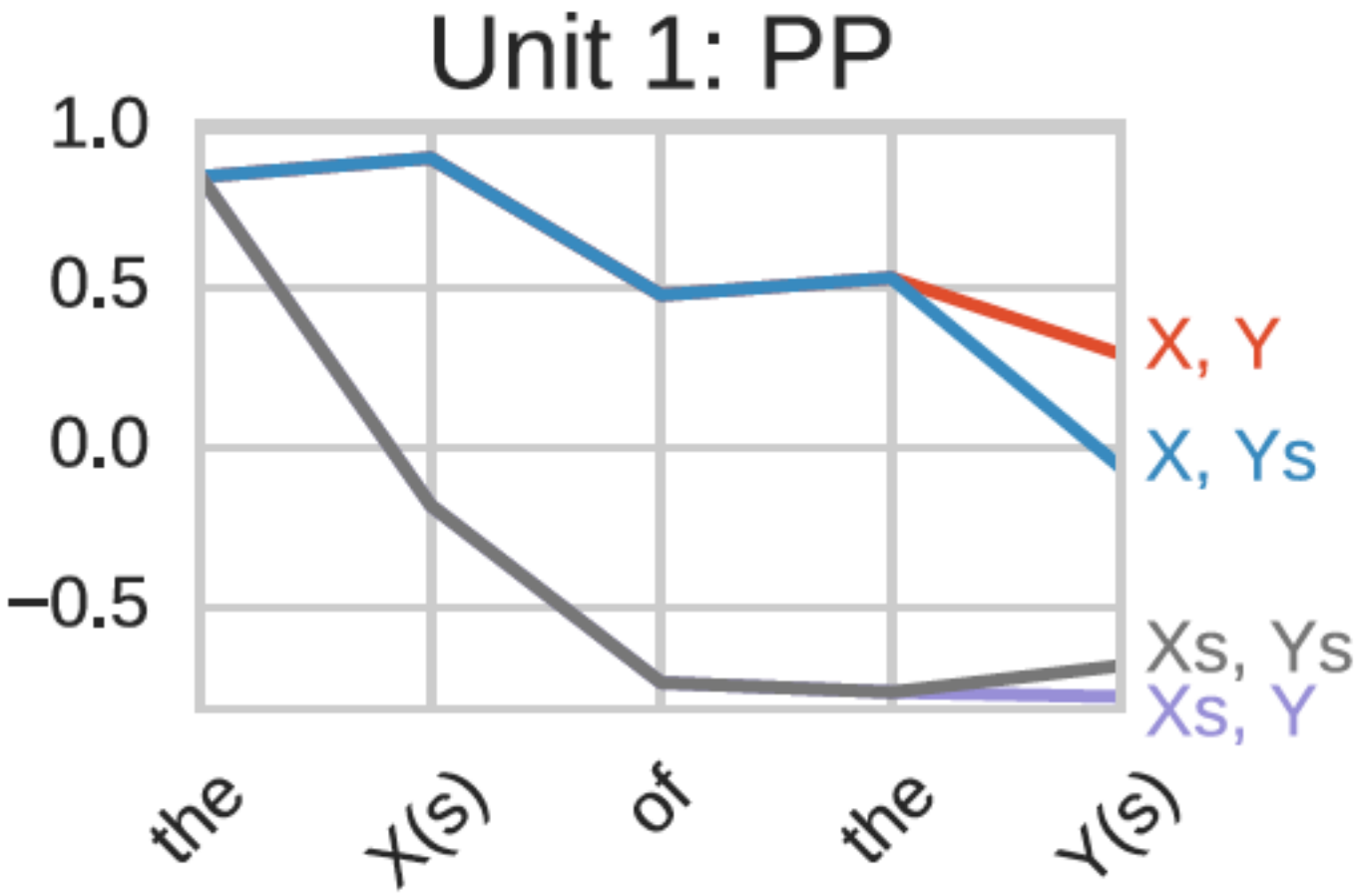
Linzen et al. (TACL 2016)

Yes, if you directly train to predict the verb form:



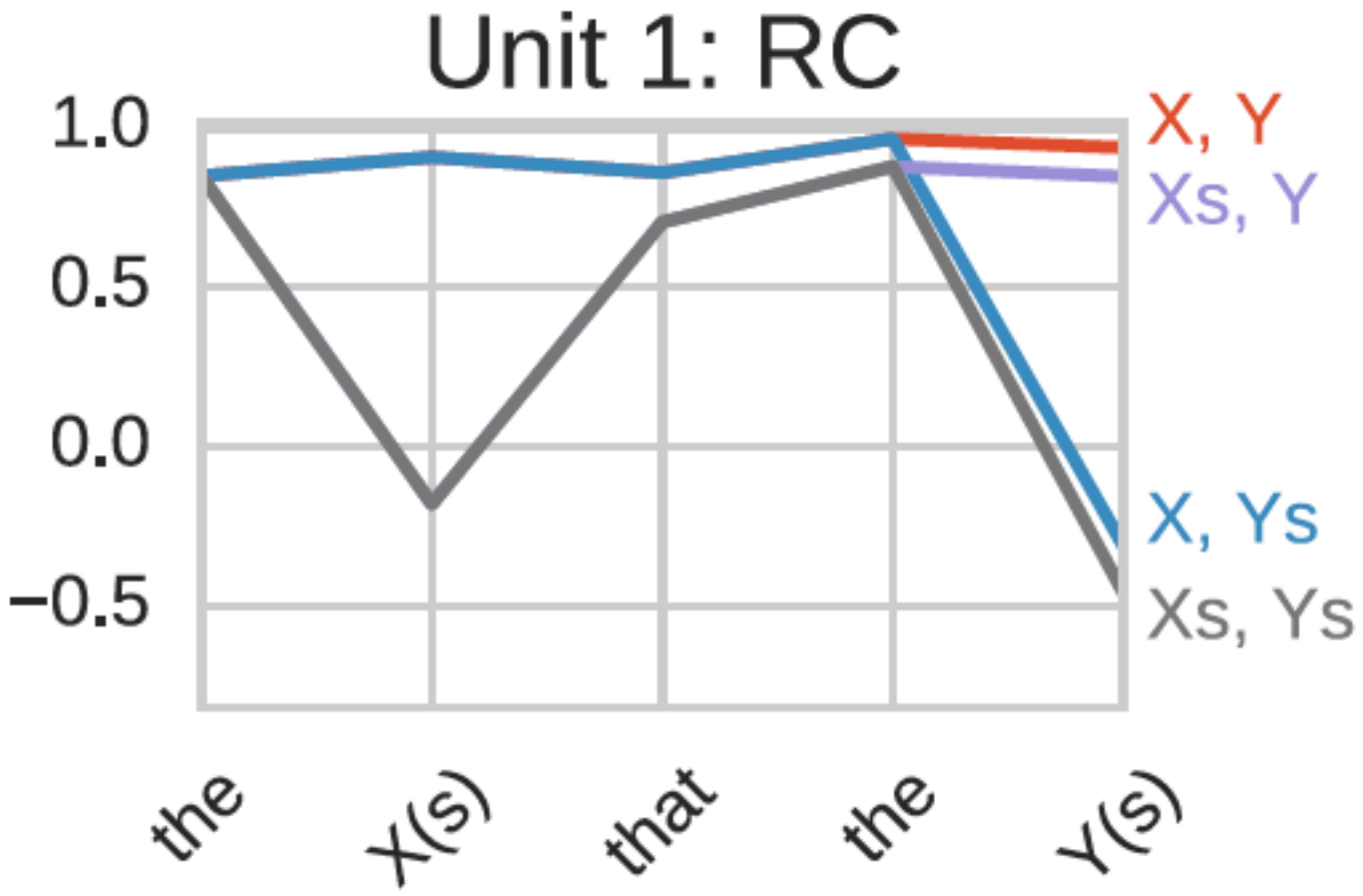
Linzen et al. (TACL 2016)

LSTM unit that remembers number (singular or plural) of earlier noun (X):



Linzen et al. (TACL 2016)

in a relative clause (RC), same LSTM unit recognizes that verb number agrees with Y:



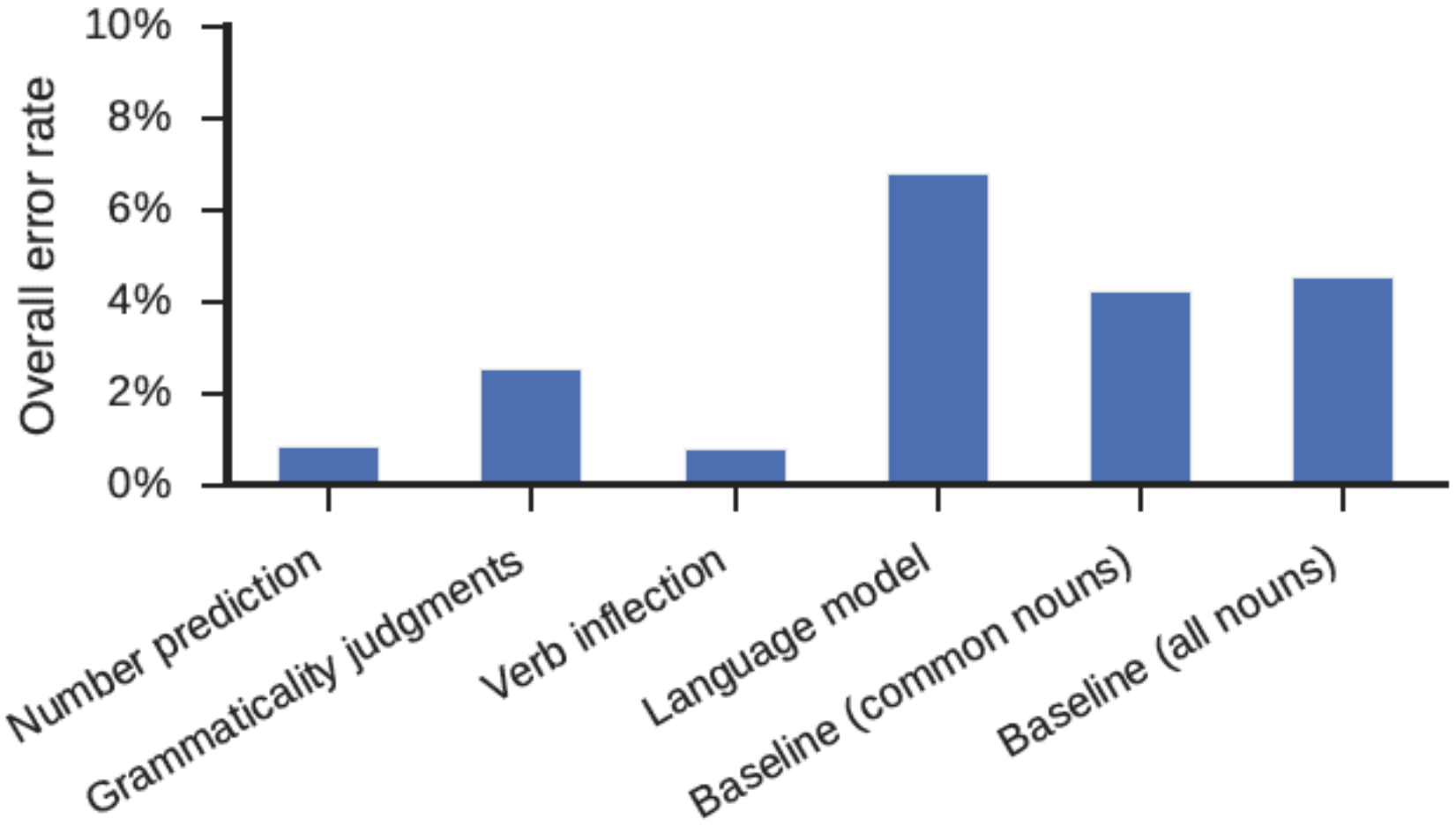
Linzen et al. (TACL 2016)

But how about if you train the LSTM as a language model?

Training objective	Sample input	Training signal	Prediction task
Number prediction	*The keys to the cabinet*	PLURAL	SINGULAR/PLURAL?
Verb inflection	*The keys to the cabinet [is/are]*	PLURAL	SINGULAR/PLURAL?
Grammaticality	*The keys to the cabinet are here.*	GRAMMATICAL	GRAMMATICAL/UNGRAMMATICAL?
Language model	*The keys to the cabinet*	are	$P(are) > P(is)$?

Linzen et al. (TACL 2016)

Then it doesn't do well at predicting agreement:



Linzen et al. (TACL 2016)