# Towards Universal Paraphrastic Sentence Embeddings

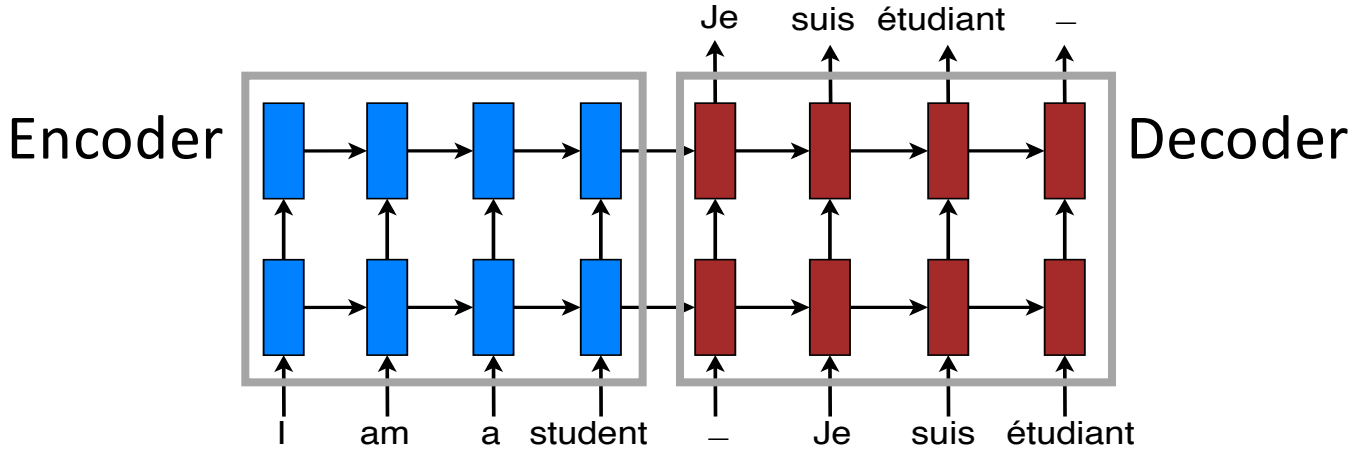

John Wieting

Joint work with Mohit Bansal, Kevin Gimpel, and Karen Livescu

# Goal

We study how to model the compositionality of natural language that is agnostic to the domain of the text.

This is important for virtually all Natural Language Processing (NLP) problems (Neural MT, QA, chat bots, etc.).




From Luong and Manning (2015)

# Goal

We focus primarily on modelling composition for the problem of semantic similarity.

Other ways are needed.
We must find other ways.

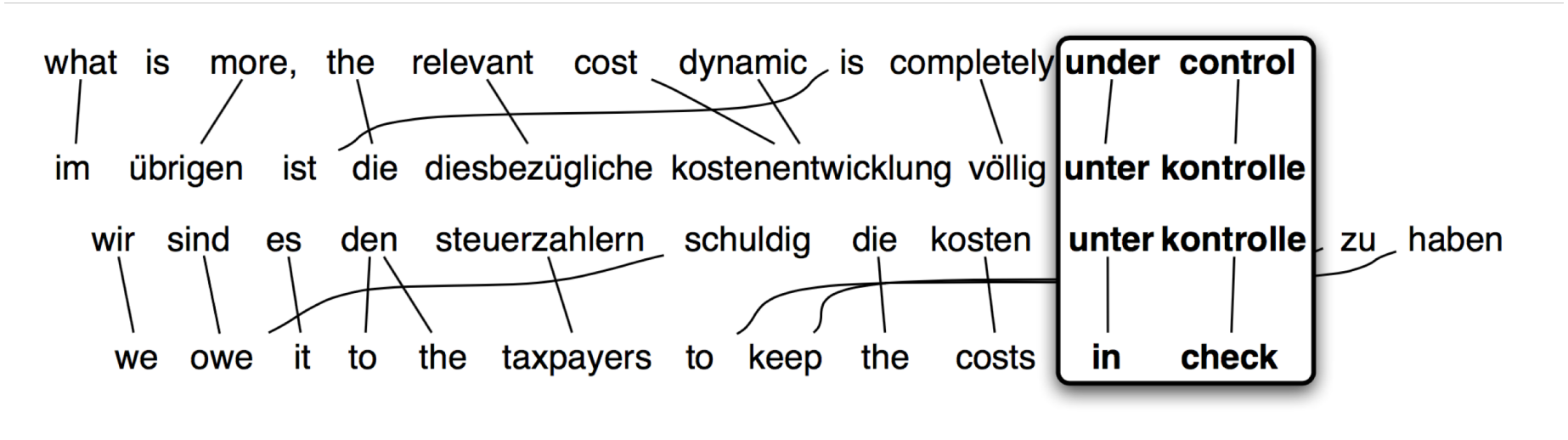
4.4

I absolutely do believe there was an iceberg in those waters.
I don't believe there was any iceberg at all anywhere near the Titanic.

1.2

# Where do we start?

Find some data.




From Bannard and Callison-Burch (2005)

# The Paraphrase Database

From Ganitkevitch, Van Durme, and Callison-Burch, 2013

be given the opportunity to
a saving
i can hardly hear you .
laying the foundations
making every effort
do better than that
…

have the possibility of
business income
you 're breaking up .
pave the way
to do its utmost
do more
…

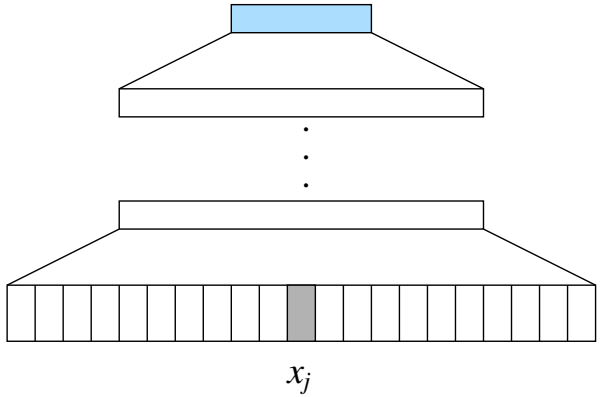
and tens of millions more!!!

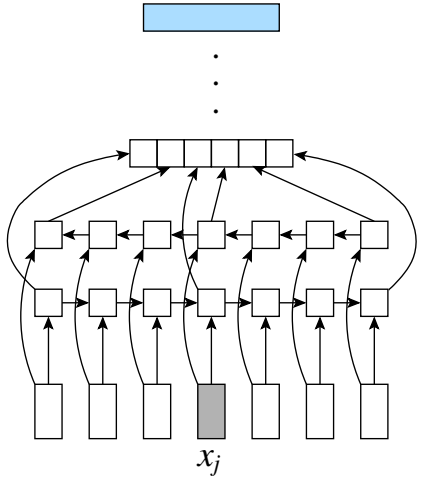
# Modelling composition

Since we want to learn representations, we need an encoder:

$g$: text sentence → fixed length vector

We experimented with 8 encoders.



# Objective function

$$\sum_{\langle x_1, x_2 \rangle \in \text{PPDB}} \max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_1))$$
$$+ \max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_2))$$

$g(x) =$ fixed length vector

$t_1 = \text{argmax}_{t:\langle \cdot, \cdot \rangle \in \text{batch}, t \neq x_1, x_2} (\cos(g(x_1), g(t))$

+ regularization!

Used separate $L_2$ regularization for word embeddings and compositional parameters

# Objective function

$$\sum_{\langle x_1, x_2 \rangle \in \text{PPDB}} \max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_1))$$

$$+\max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_2))$$

sums over pairs in Paraphrase Database

# Objective function

$$\sum_{\langle x_1, x_2 \rangle \in \text{PPDB}} \max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_1))$$

$$+\max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_2))$$

sums over pairs in Paraphrase Database

cosine similarity of phrases in positive example

# Objective function

$$\sum_{\langle x_1, x_2 \rangle \in \text{PPDB}} \max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_1))$$

$$+\max(0, \delta - \cos(g(x_1), g(x_2)) + \cos(g(x_1), g(t_2))$$

sums over pairs in Paraphrase Database

cosine similarity of phrases in positive example

cosine similarity of phrases in negative examples

# Evaluation

We evaluate on 22 out-of-domain datasets and 2 in-domain.

For model selection, only use an in-domain dataset.

Domains of the 22 datasets include:

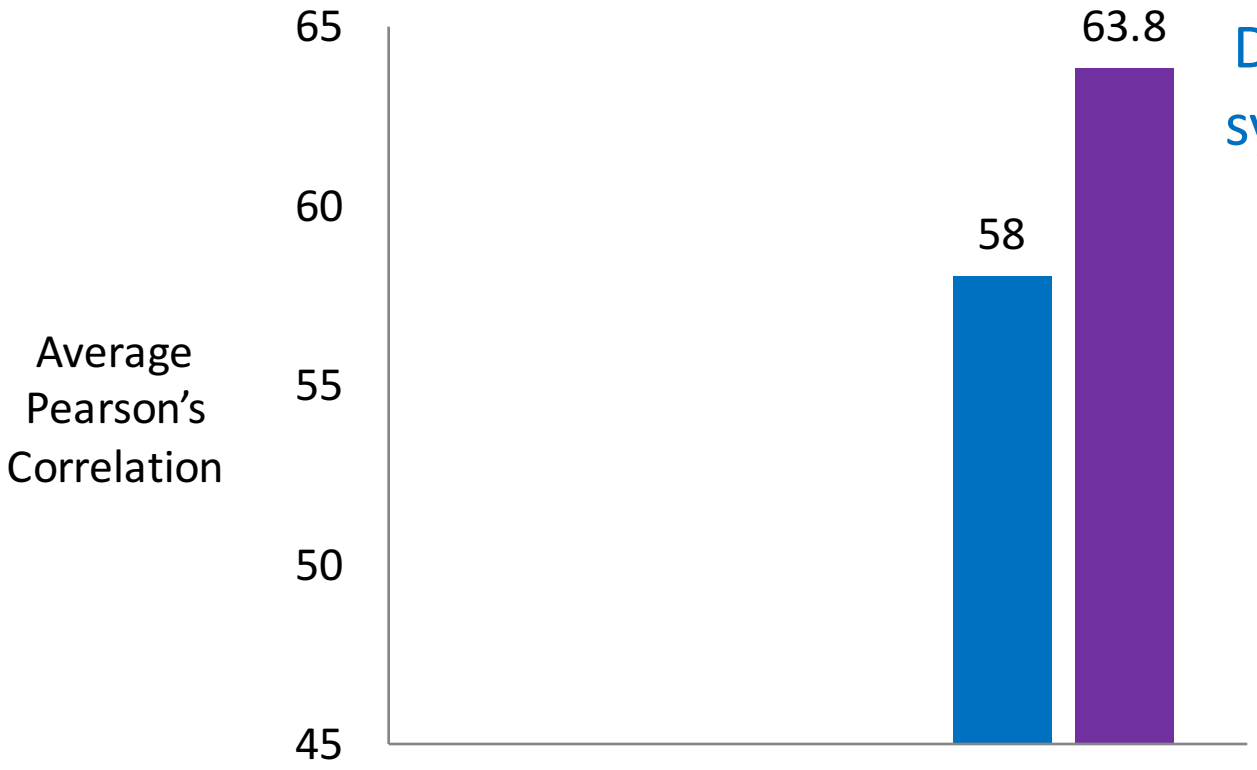
- web forum posts
- tweets
- MT output
- news
- headlines
- glosses
- image and video captions
- ….

# In-domain datasets

A sample of PPDB, annotated by Turkers. We compare with two datasets, from Wieting et al. (2015) and Pavlick et al. (2015).

| | | |
|---|---|---|
| can not be separated from | is inseparable from | 5.0 |
| hoped to be able to | looked forward to | 3.4 |
| come on , think about it | people , please | 2.2 |
| how do you mean that | what worst feelings | 1.6 |

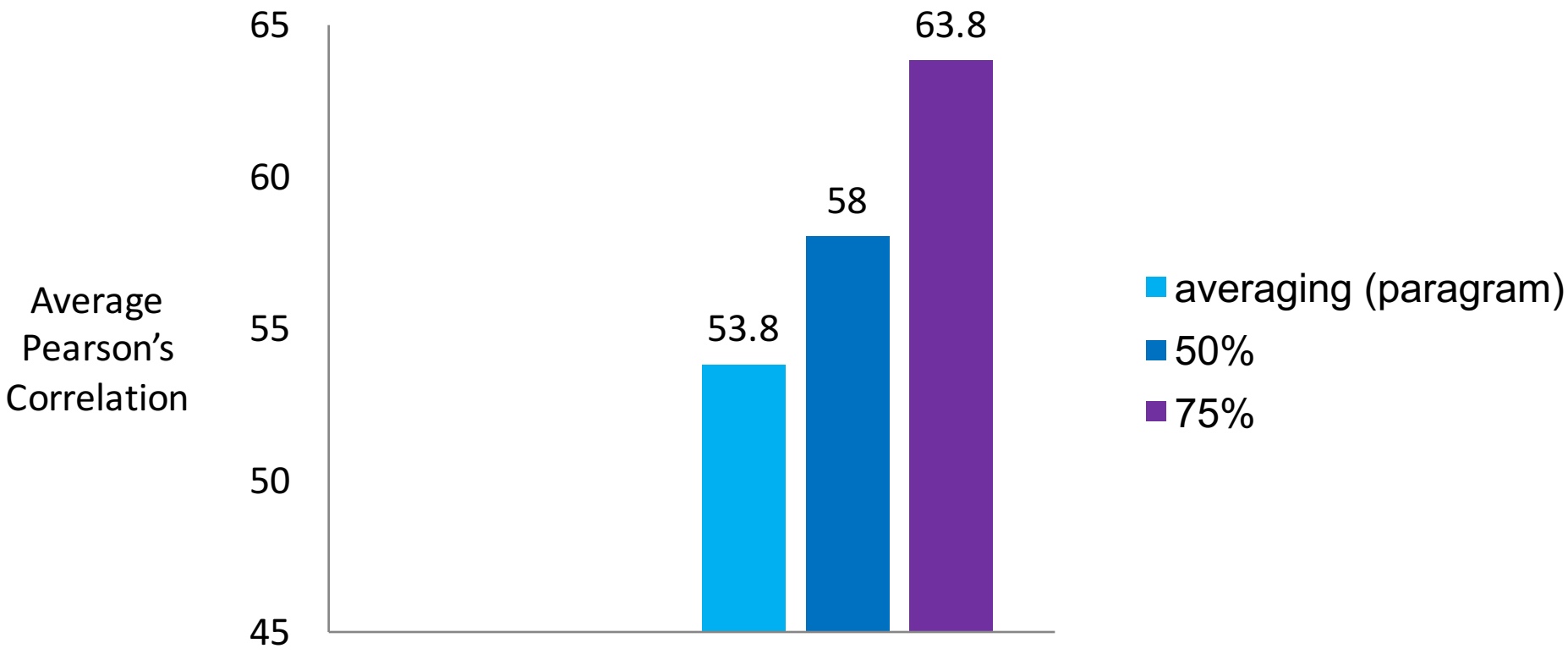
# Average Pearson's correlation on 22 datasets



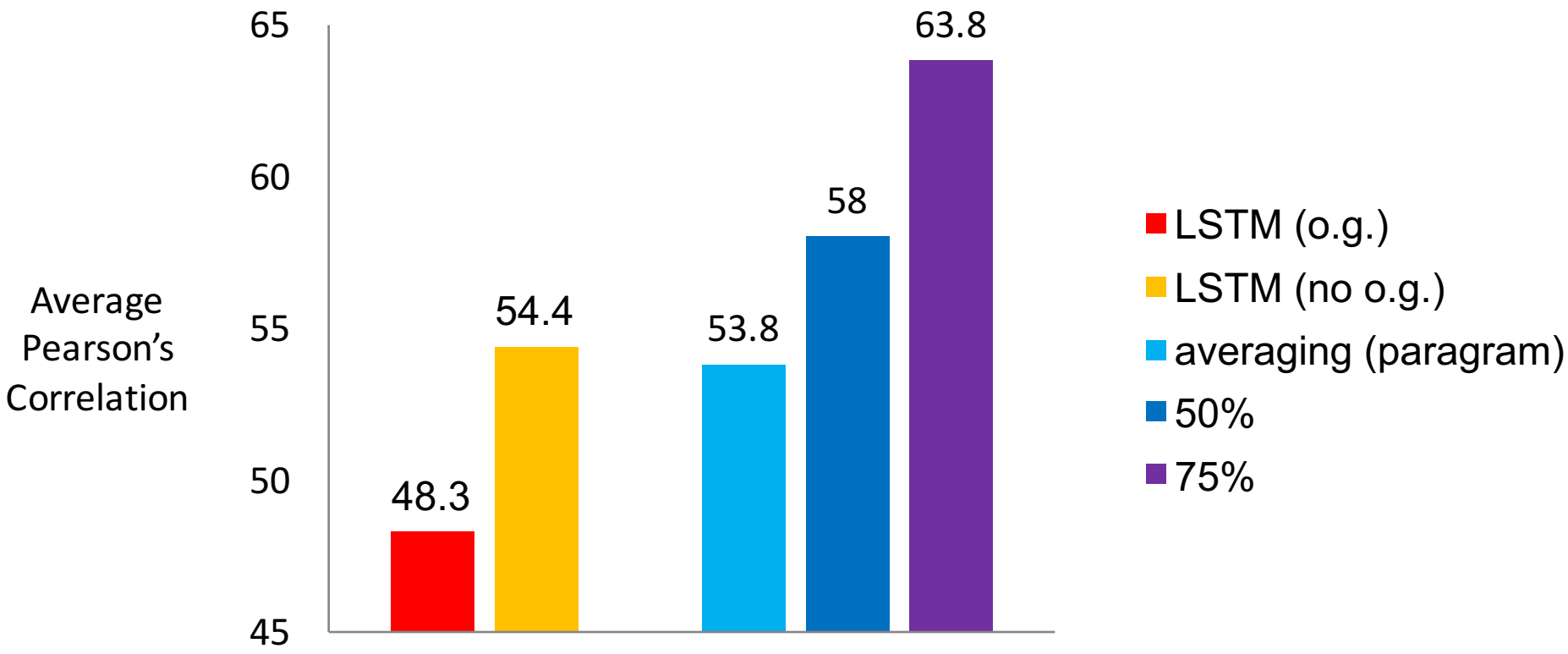

Depending on task, anywhere from 26-89 systems were submitted which had access to training data and external resources.

Also tried using skip-thought vectors and averaging GloVe embeddings, and they were not stronger than paragram-sl999.

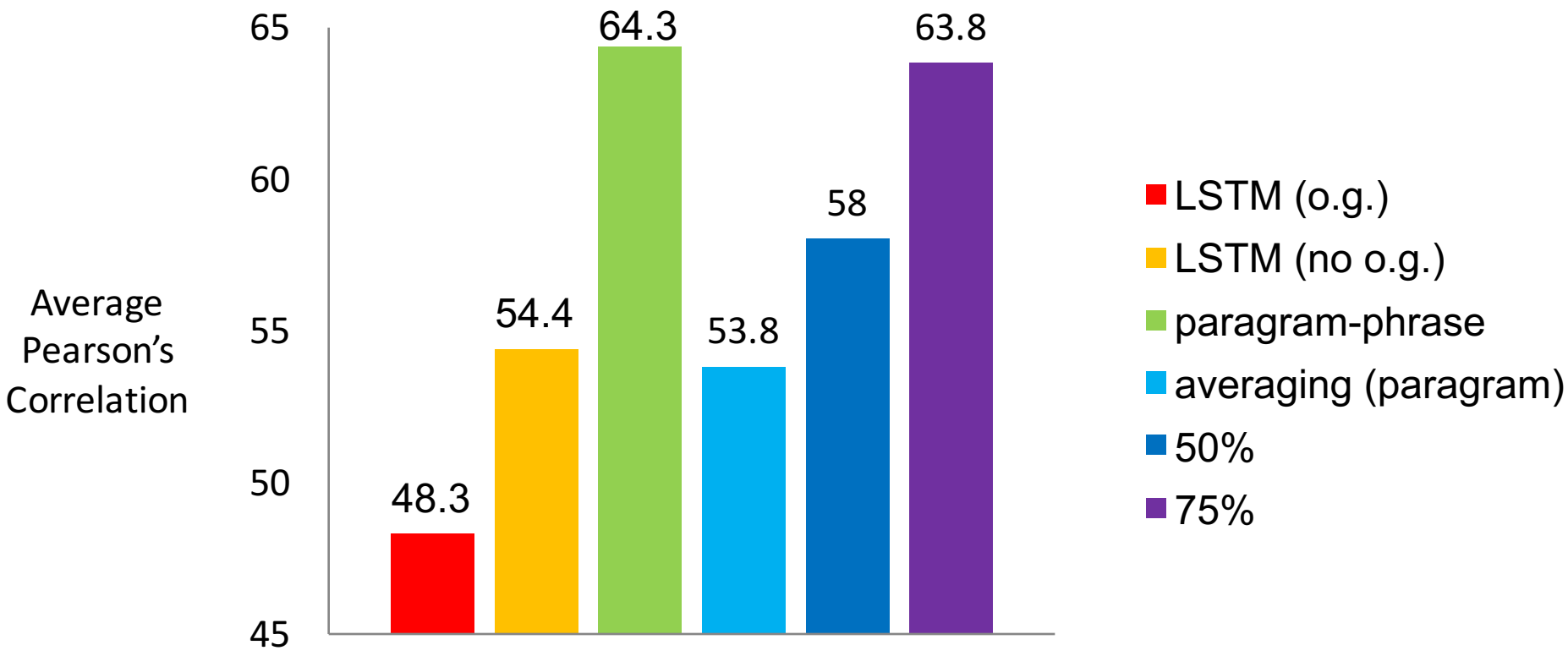
# Average Pearson’s correlation on 22 datasets



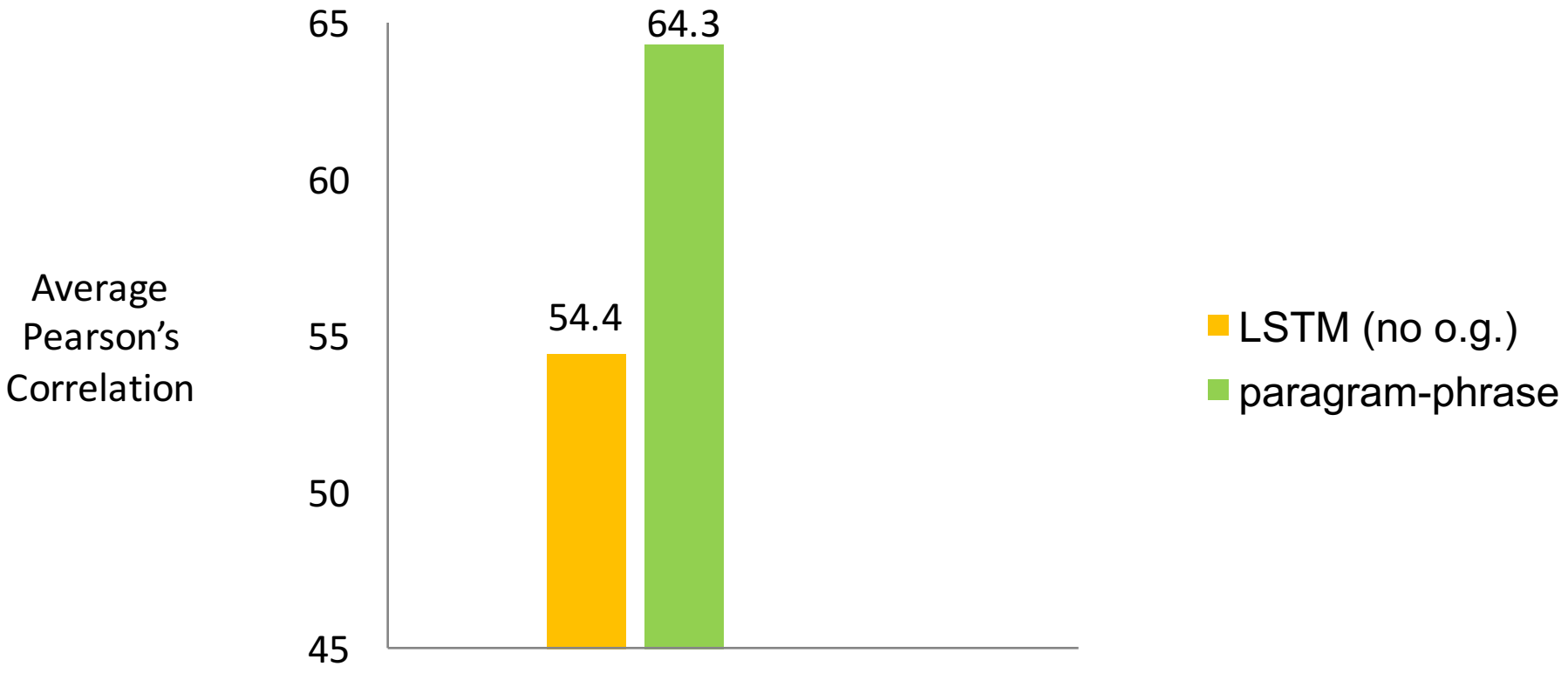
# Average Pearson's correlation on 22 datasets



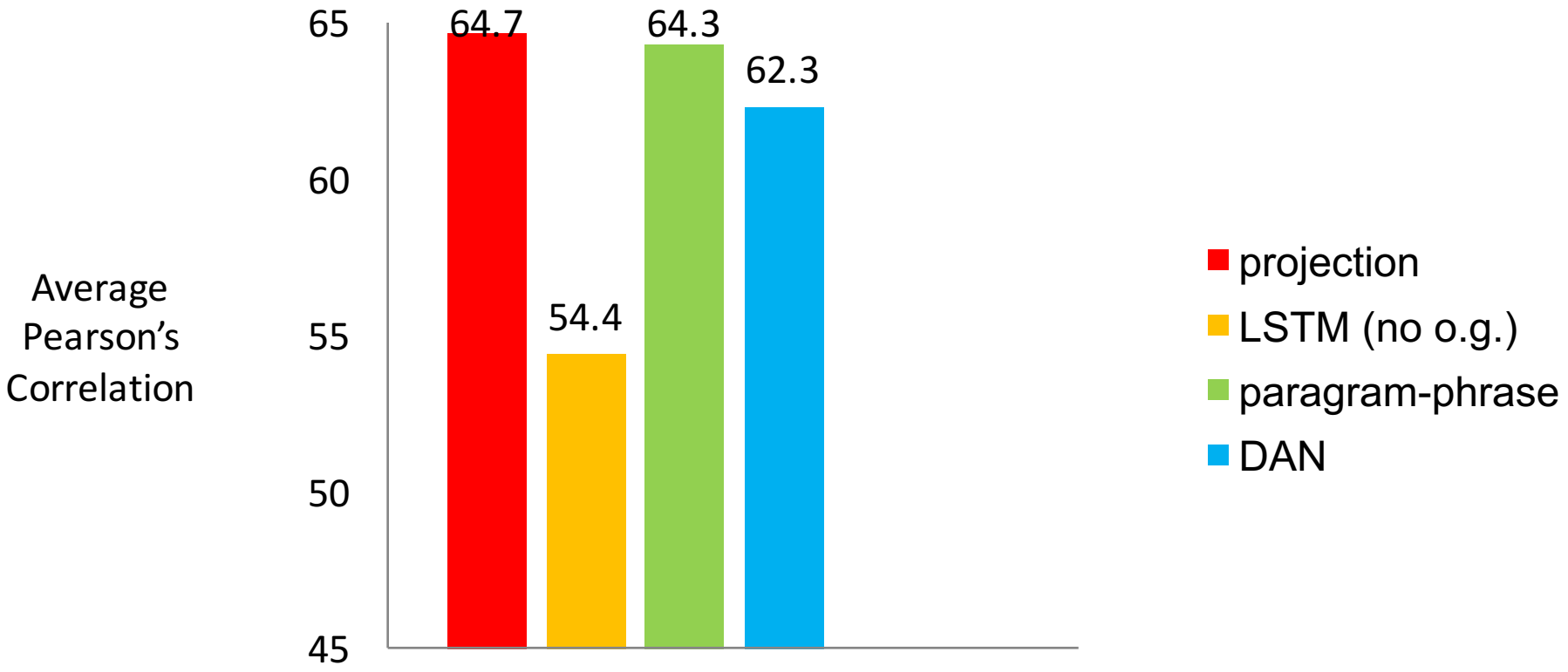
# Average Pearson's correlation on 22 datasets



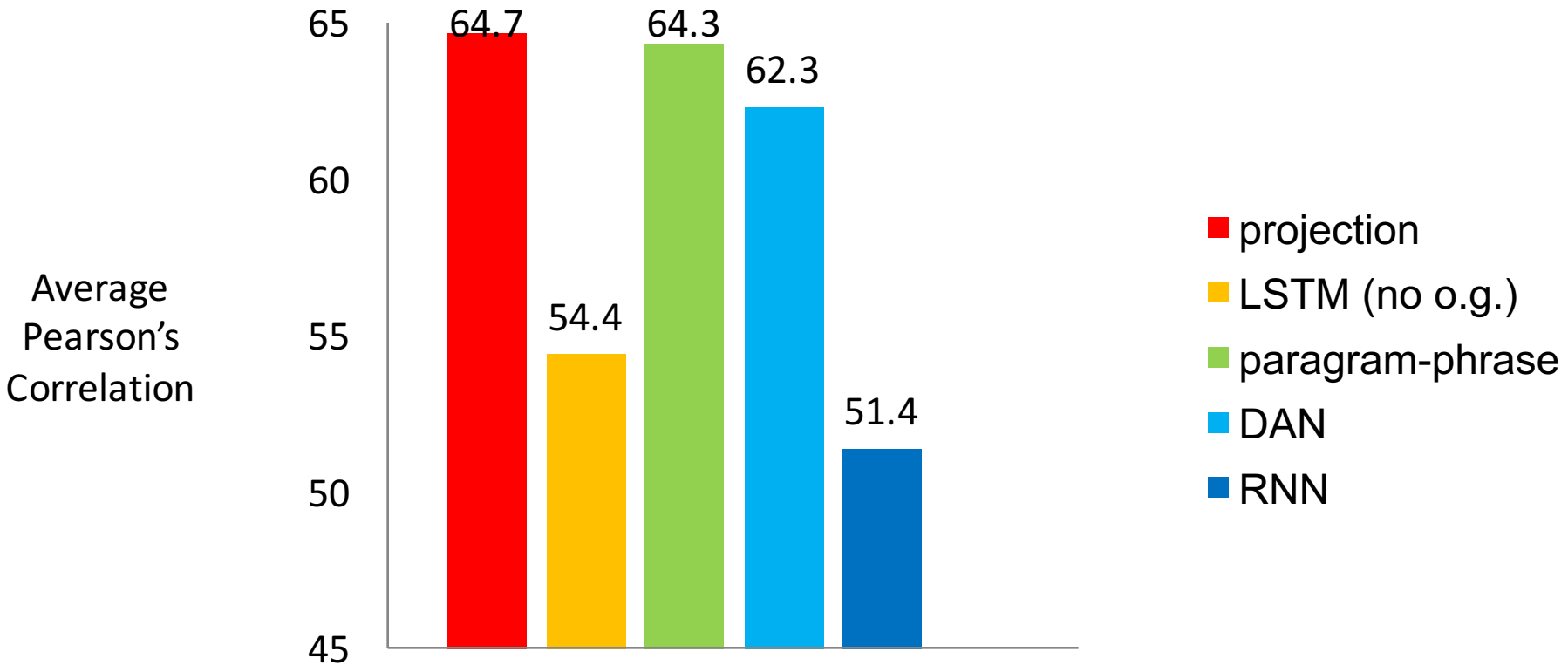
# Average Pearson's correlation on 22 datasets



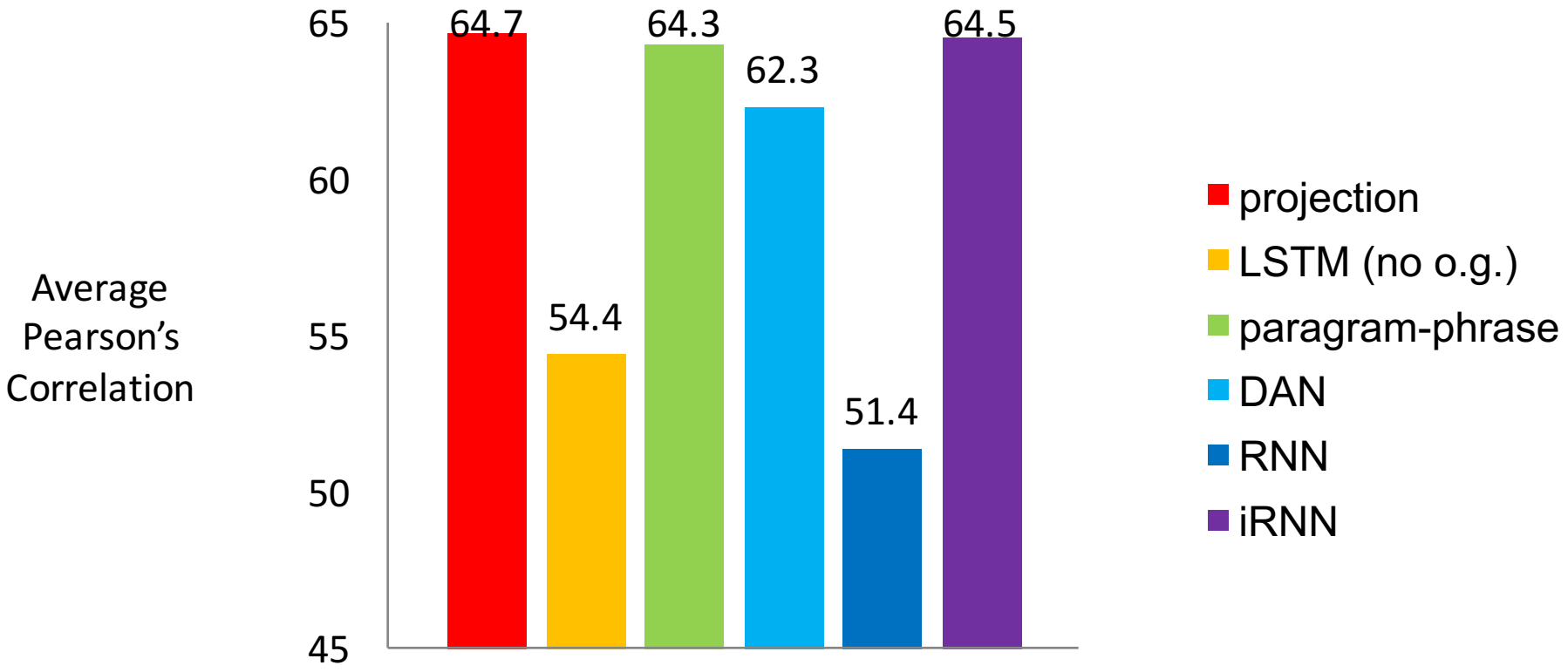
# Average Pearson's correlation on 22 datasets



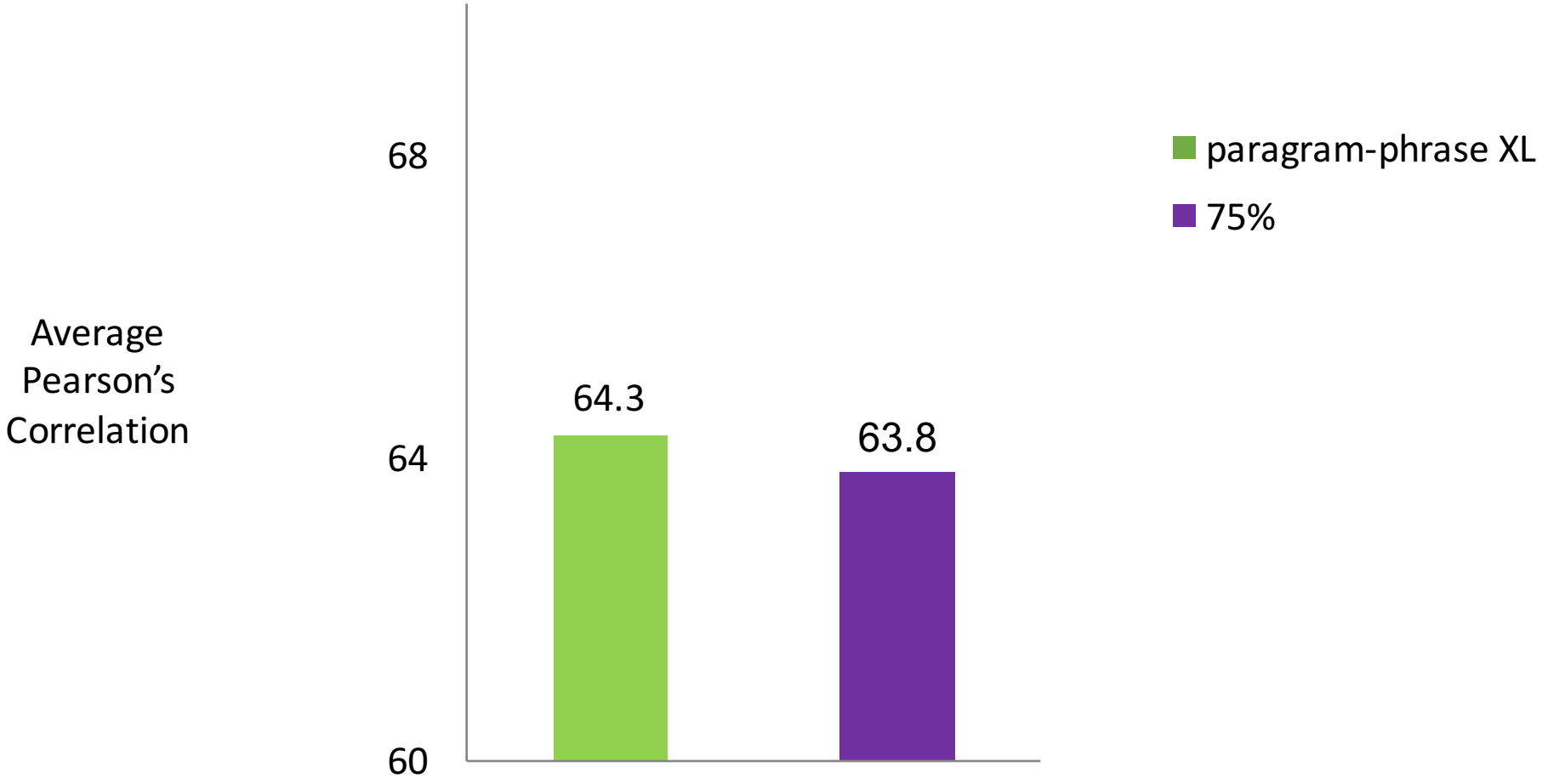
# Average Pearson's correlation on 22 datasets



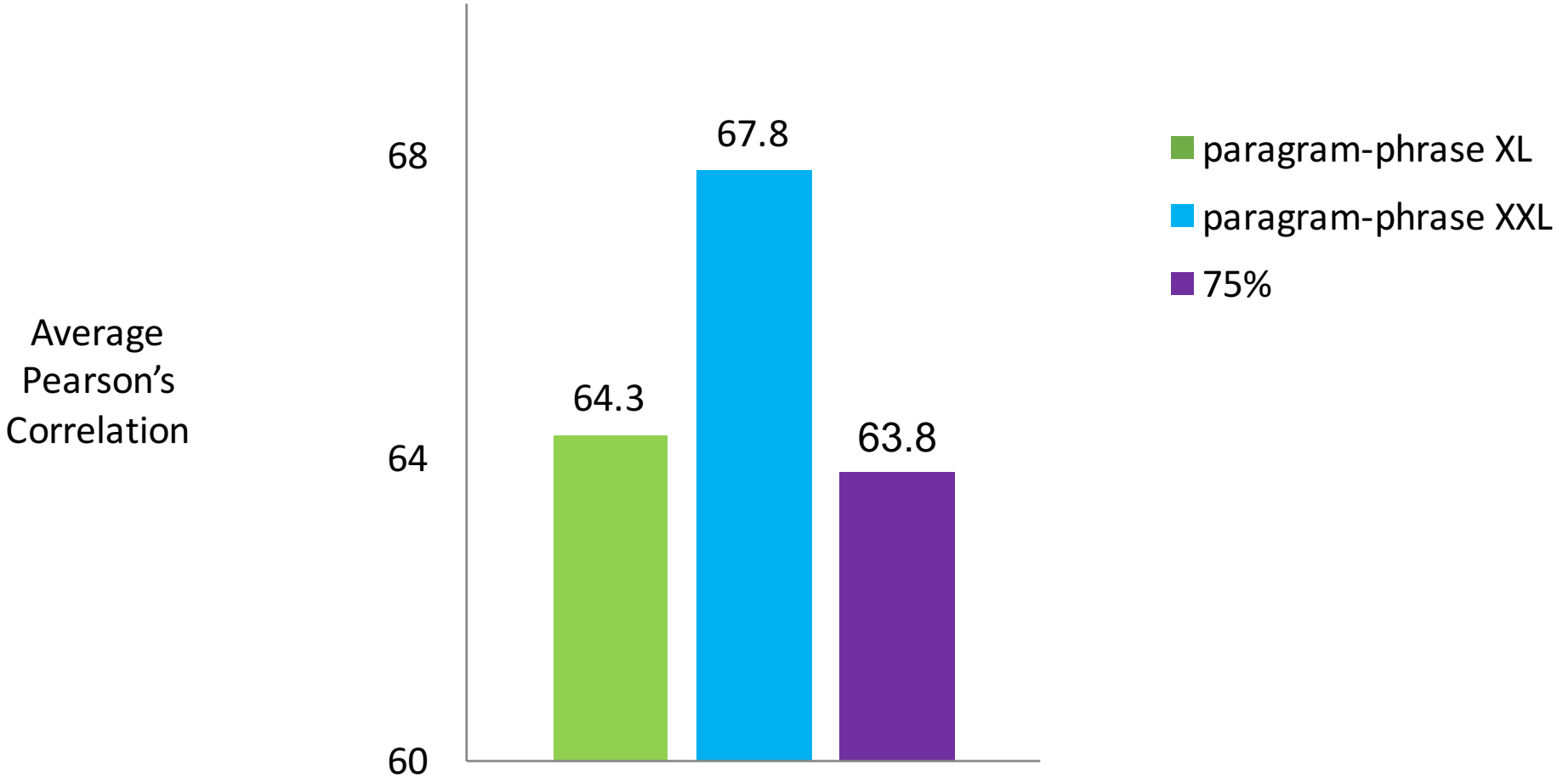
# Average Pearson's correlation on 22 datasets



# Scaling up


Average Pearson's Correlation
68
64
60
64.3
63.8
paragram-phrase XL
75%

# Scaling up


Average Pearson's Correlation
68
64
60
64.3
67.8
63.8
paragram-phrase XL
paragram-phrase XXL
75%

# Reflection

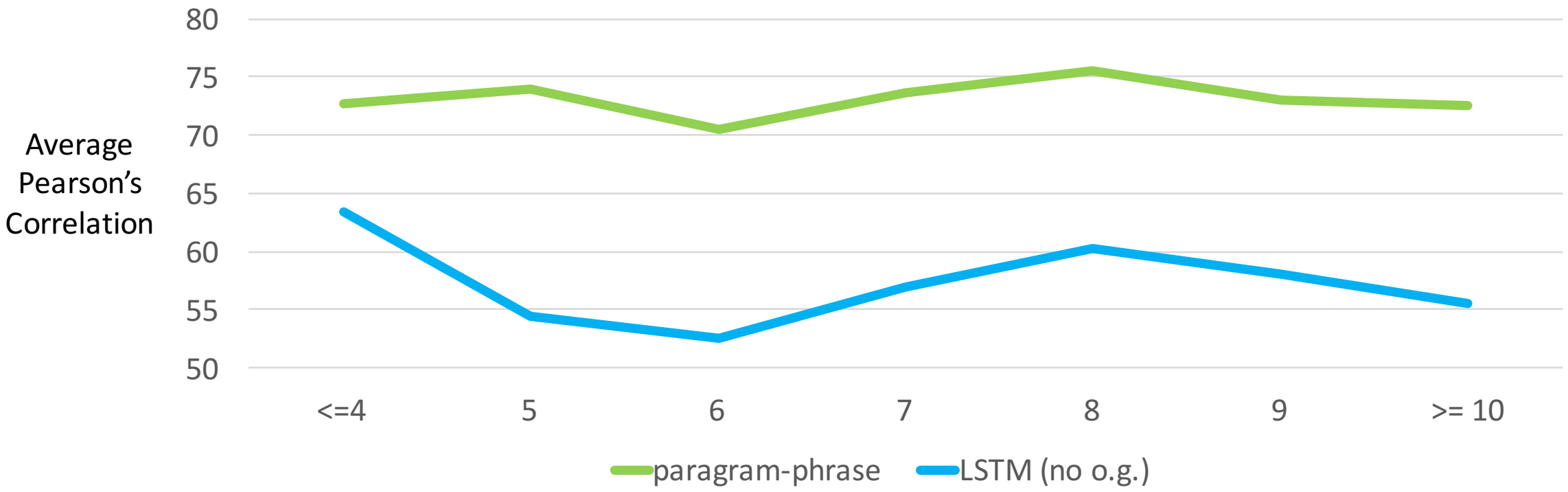
Why did the LSTM do worse?

Does it only do well on short sentences?

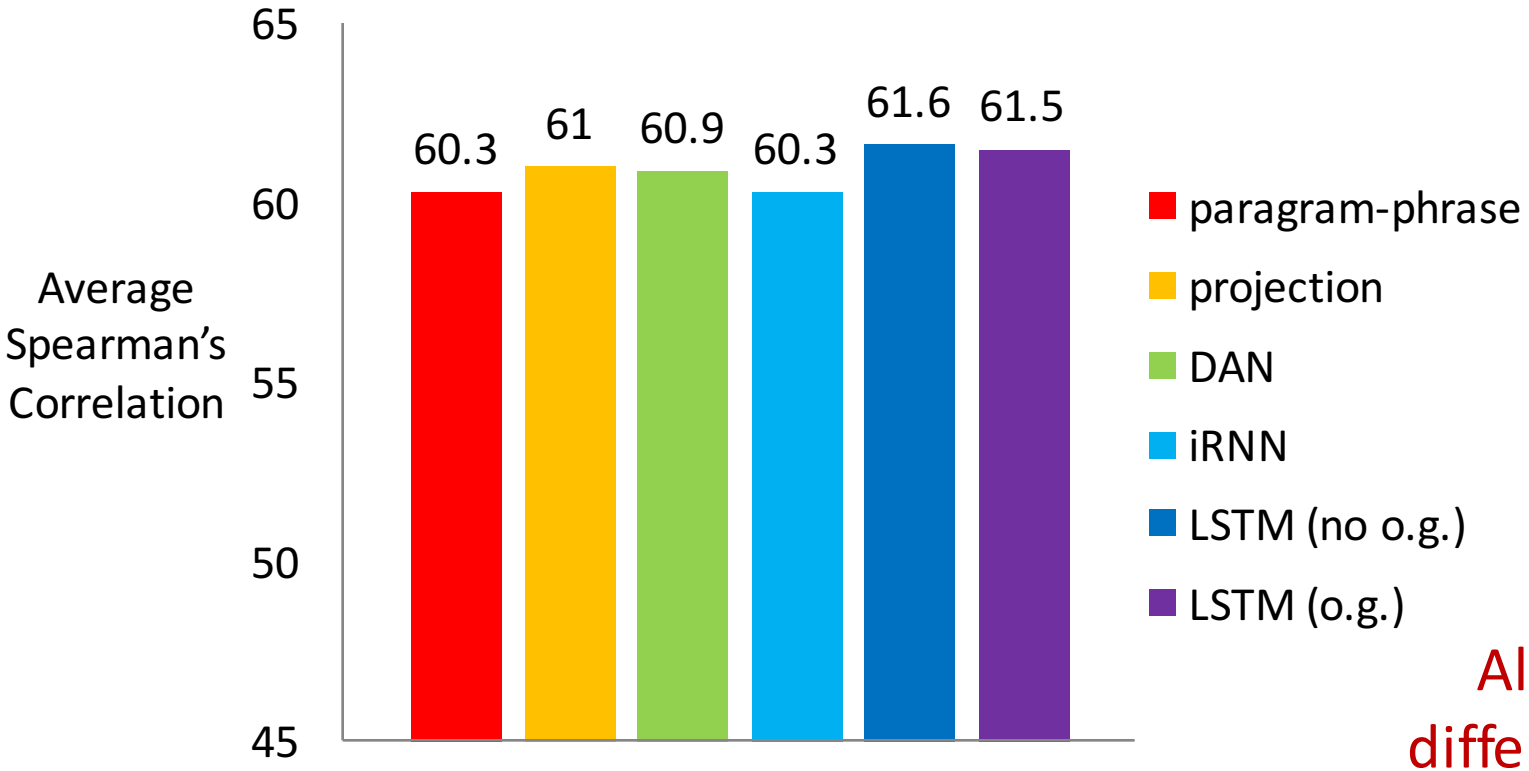
Did it overfit to the in-domain task?

Was there insufficient parameter tuning?

# Length


Average Pearson's Correlation
80
75
70
65
60
55
50
<=4
5
6
7
8
9
>= 10
paragram-phrase
LSTM (no o.g.)

# Overfitting on in-domain data



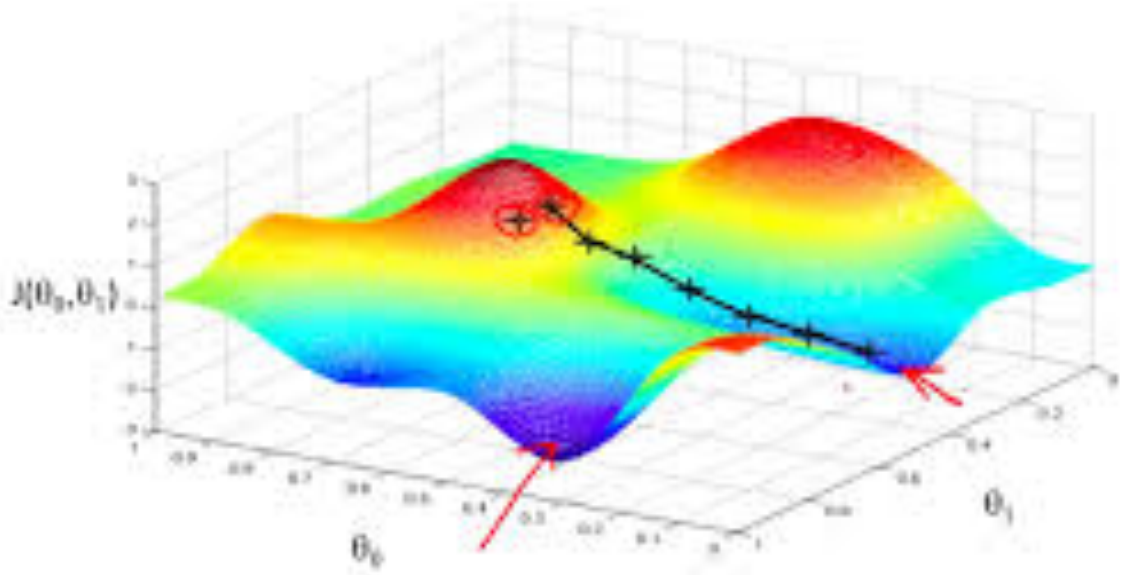

Also investigated average difference between cosine sim of positive and negative examples

# Parameter tuning

Hard to show a negative result, but we did a lot of experiments to:

explore hyperparameter space of each model

reduce potential optimization issues



# Parameter tuning

Tuned:

- optimizer (Adagrad or Adam)
- gradient clipping
- learning rate
- batch-size
- $\lambda_c, \lambda_w$
- $\delta$
- type of sampling
- activation function, number of layers (if applicable)

# Other use cases?

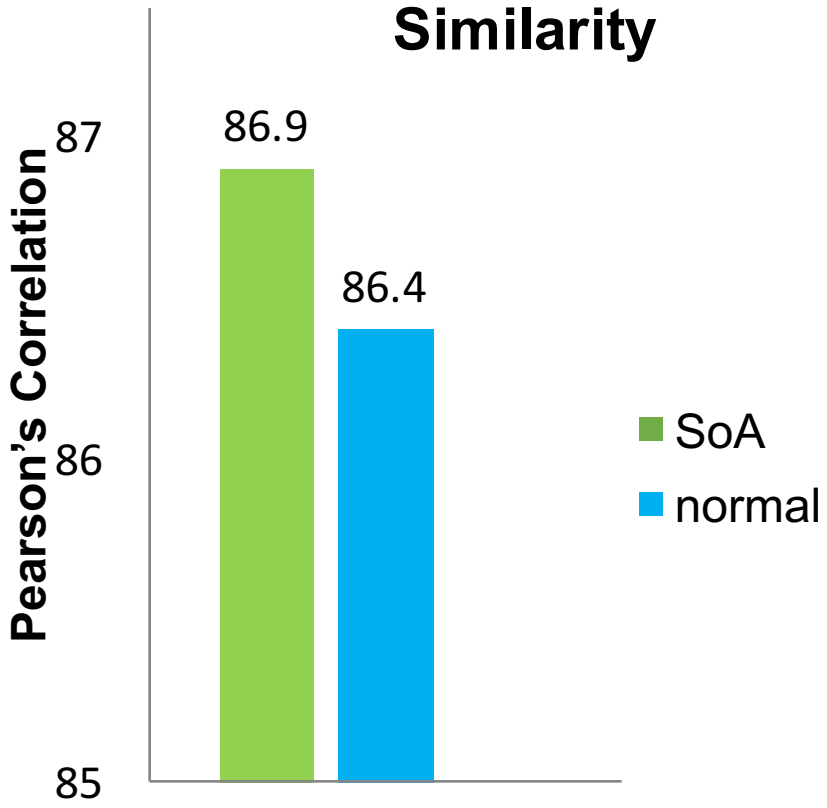
Yes!

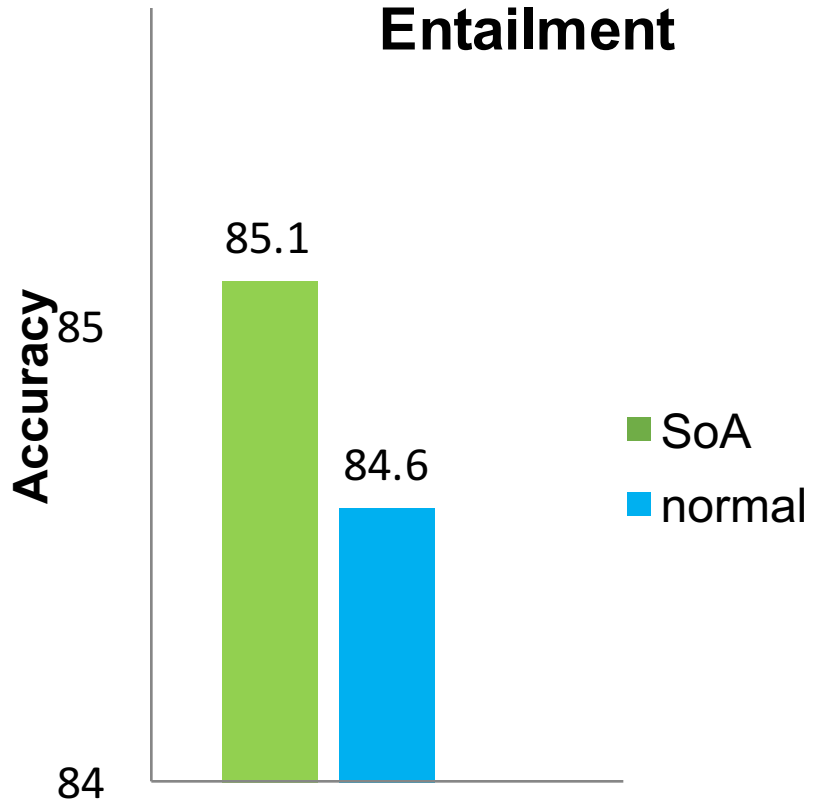
Can improve specific similarity/entailment tasks when used to initialize/regularize other models.

Can be used as features for at least similarity and entailment tasks.

# Initialization/Regularization

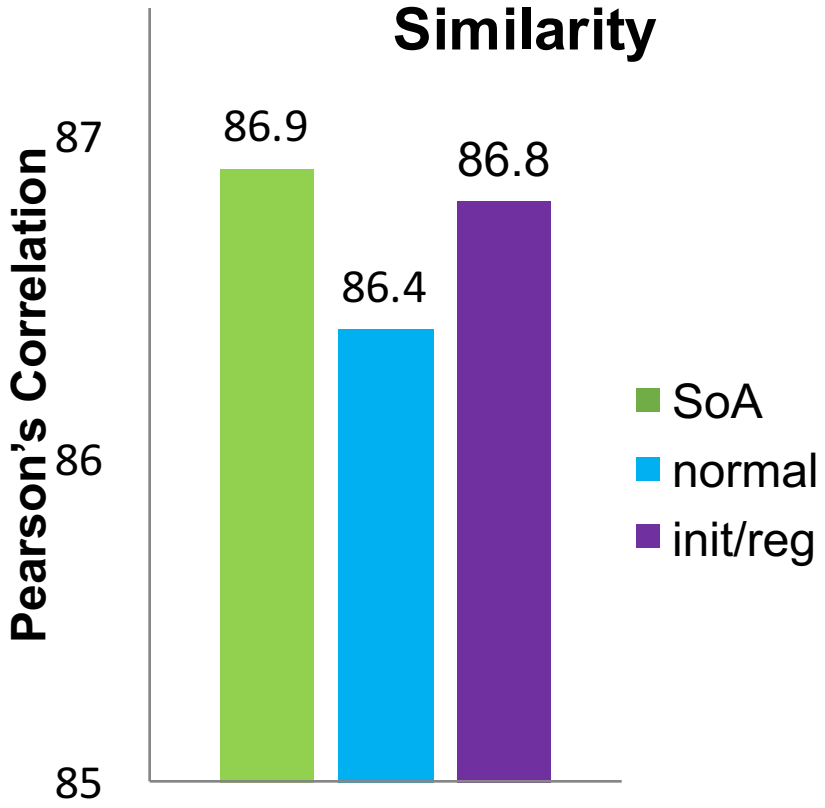
word-averaging

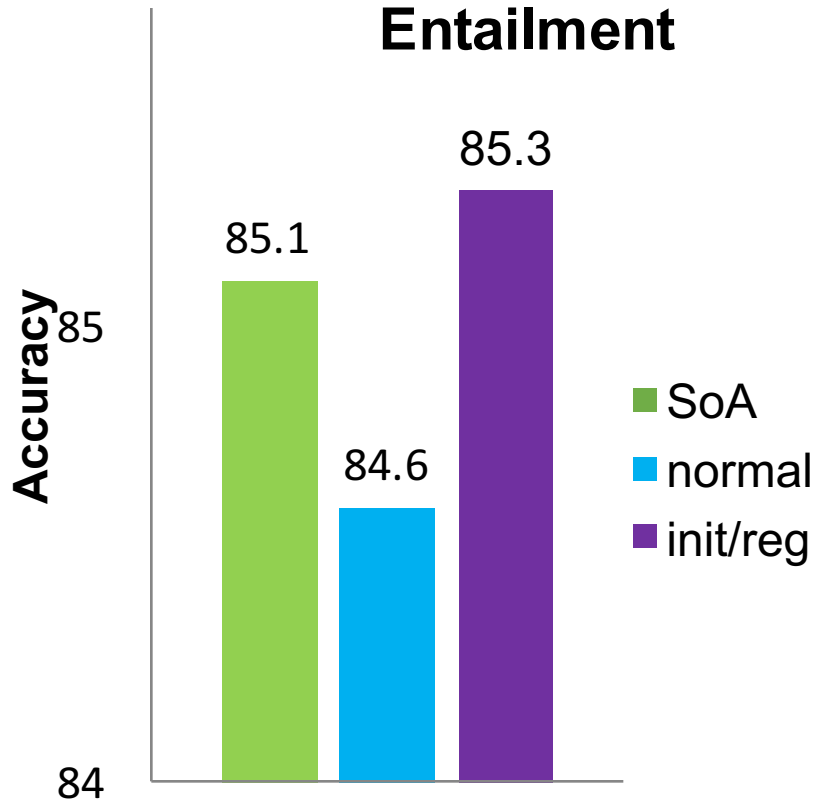


# Initialization/Regularization

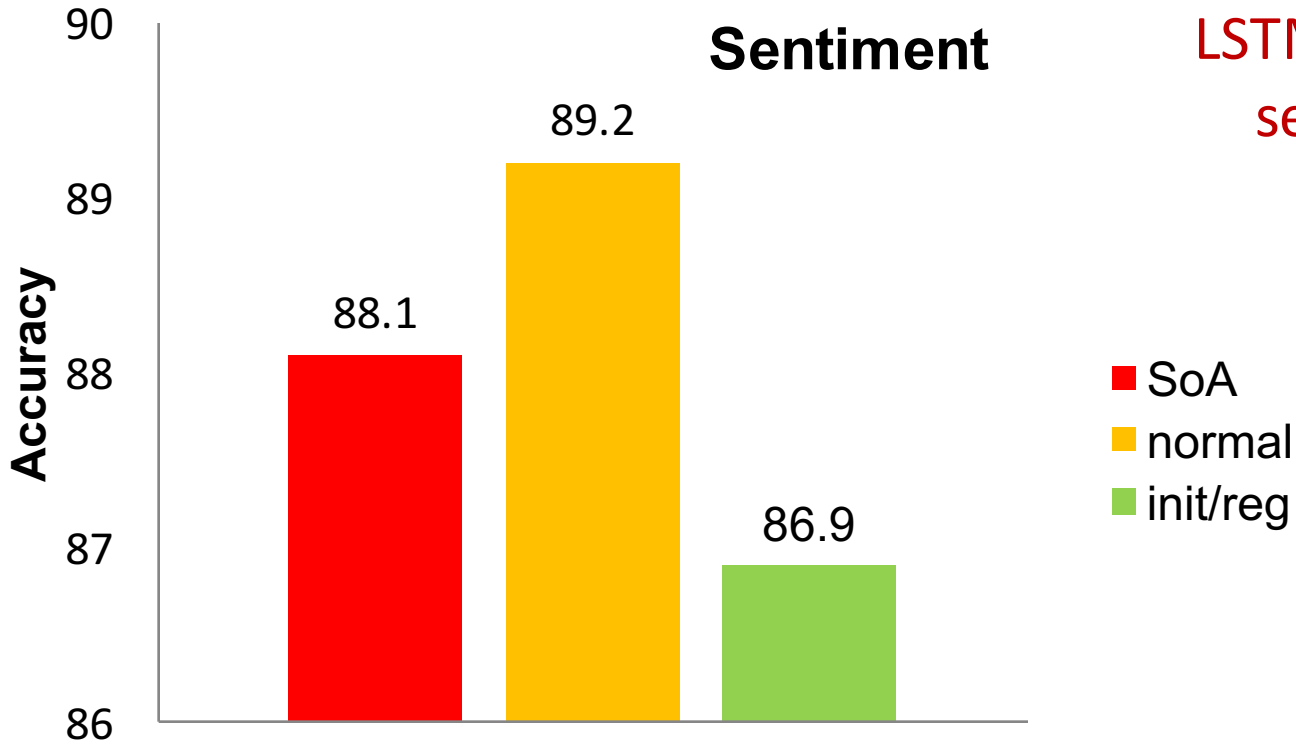
word-averaging



# Initialization/Regularization

LSTM (o.g.)




LSTMs perform really well on sentiment classification!

LSTM sentence models in our transfer learning setting perform poorly, so this result isn't too surprising.

# Qualitative/Quantitative analysis

Found that a significant part of the power of our embeddings is due to re-weighting $L_2$ norms of words by their importance (i.e. *18* versus *of*)

# Qualitative/Quantitative analysis

Found that a significant part of the power of our embeddings is due to re-weighting $L_2$ norms of words by their importance (i.e. *18* versus *of*)

| | paragram-phrase | paragram-simlex |
|---|---|---|
| unlike | contrary, contrast, opposite | than, although, whilst |
| lookin | staring, looking, watching | doin, goin, talkin |
| disagree | agree, concur, agreeing | disagreement, differ, dispute |

# Qualitative/Quantitative analysis

Found that a significant part of the power of our embeddings is due to re-weighting $L_2$ norms of words by their importance (i.e. *18* versus *of*)

| | paragram-phrase | paragram-simlex |
|---|---|---|
| unlike | contrary, contrast, opposite | than, although, whilst |
| lookin | staring, looking, watching | doin, goin, talkin |
| disagree | agree, concur, agreeing | disagreement, differ, dispute |

Spearman's correlation of -45.1 between performance and OOV %.

# Qualitative/Quantitative analysis

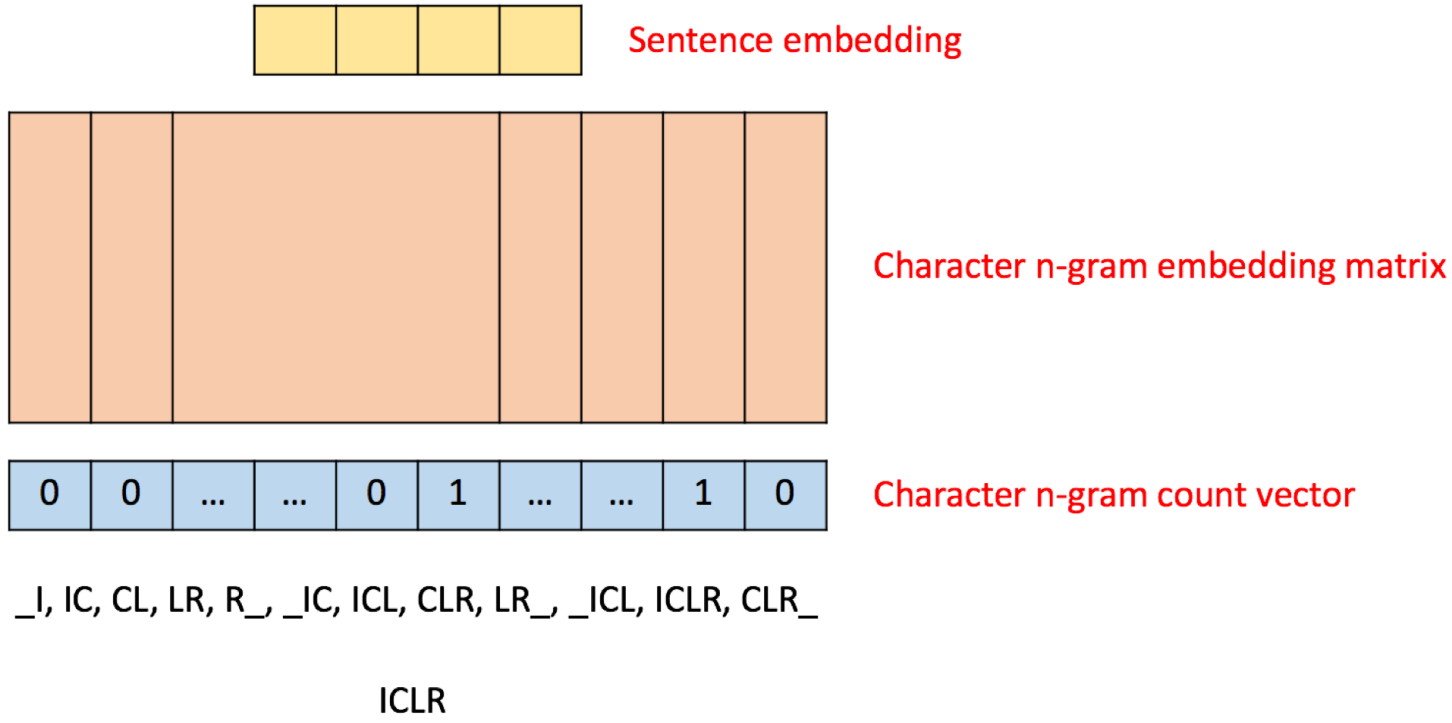
Found that a significant part of the power of our embeddings is due to re-weighting $L_2$ norms of words by their importance (i.e. *18* versus *of*)

| | paragram-phrase | paragram-simlex |
|---|---|---|
| unlike | contrary, contrast, opposite | than, although, whilst |
| lookin | staring, looking, watching | doin, goin, talkin |
| disagree | agree, concur, agreeing | disagreement, differ, dispute |

Spearman's correlation of -45.1 between performance and OOV %.

# Character n-gram model

Inspired by the Deep Structured Semantic Model or Deep Semantic Similarity Model (MSR, 2013-2016)



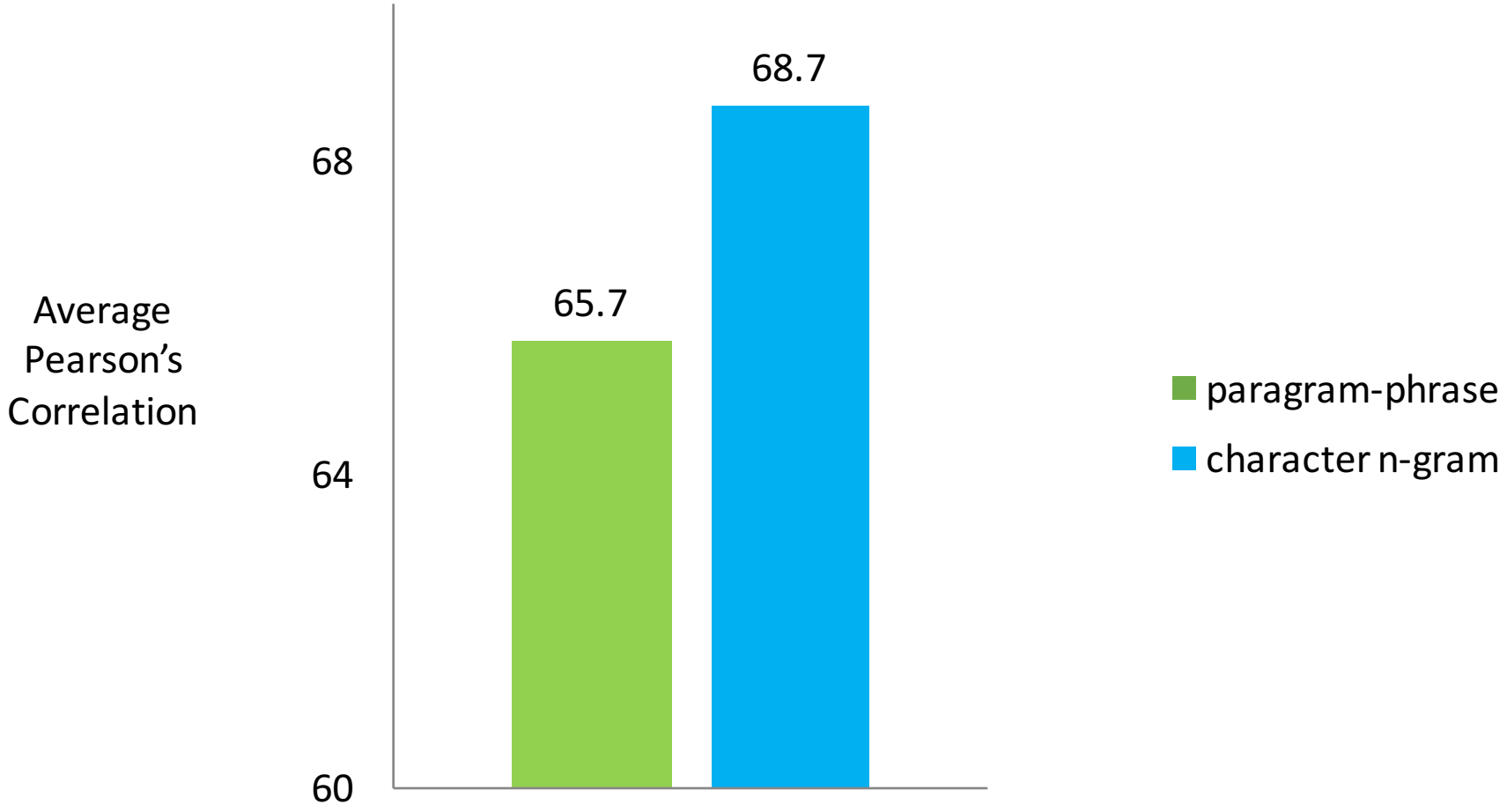
# Character n-gram model

Able to model very rare words, context, and still generalizes nicely!

| | character n-gram embeddings | paragram-phrase |
|---|---|---|
| not capable | incapable, unable, incapacity | not, capable, stalled |
| not possible | impossible, impracticable, unable | not, stalled, possible |
| not sufficient | insufficient, sufficient, inadequate | not, sufficient, stalled |

| | character n-gram embeddings |
|---|---|
| babyyyyyy | babyyyyyyy, baby, babys, babe, baby.i, babydoll, babycake, darling |
| vehicals | vehical, vehicles, vehicels, vehicular, cars, vehicle, automobiles, car |
| huge | enormous, tremendous, large, big, vast, overwhelming, immense, giant |

# Character n-gram model



# Conclusion

We have shown how, essentially using just using bilingual text, it is possible to create a strong model of composition that is not tied to a specific dataset and is both fast and easy to use.

We also raise some questions about LSTMs. Why did they not work as well in this setting? Hopefully this work can lead to even better compositional architectures that generalize across many domains.

We release code, trained models and resources to replicate and build upon our models.

# Conclusion

We have shown how, essentially using just using bilingual text, it is possible to create a strong model of composition that is not tied to a specific dataset and is both fast and easy to use.

We also raise some questions about LSTMs. Why did they not work as well in this setting? Hopefully this work can lead to even better compositional architectures that generalize across many domains.

We release code, trained models and resources to replicate and build upon our models.

Thank You!