Learning Generic Sentence Representations Using Convolutional Neural Networks

Presenter: Zhe Gan

Joint work with: Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, Lawrence Carin

Duke University & Microsoft Research

September 11th, 2017

Outline

Background

- Deep neural nets have achieved great success in learning *task-dependent* sentence representations
  - feedforward neural nets
  - recurrent neural nets
  - convolutional neural nets
  - recursive neural nets
  - ...
- Downstream tasks:
  - classification, entailment, semantic relatedness, paraphrase detection, ranking ...
- Potential drawback:
  - They are trained specifically for a certain task, requiring retraining a new model for each individual task.

Problem of interest

- Problem of interest: learning *generic* sentence representations that can be used across domains.
- In computer vision, CNN trained on ImageNet, C3D trained on Sports-1M have been used to learn a generic image/video encoder that can be transferred to other tasks.
- *How to achieve it in NLP?*
  - what dataset to use?
  - what neural net encoder to use?
  - what task to perform?
- Follow the *Skip-Thought* vector work[1]

[1] Kiros, Ryan, et al. "Skip-thought vectors" NIPS, 2015.

Review: skip-thought vectors

- Model: GRU-GRU encoder-decoder framework
- Task: Encode a sentence to predict its neighboring two sentences
- Dataset: BookCorpus, 70M sentences over 7000 books
- Input: *I got back home. I could see the cat on the steps. This was strange.*

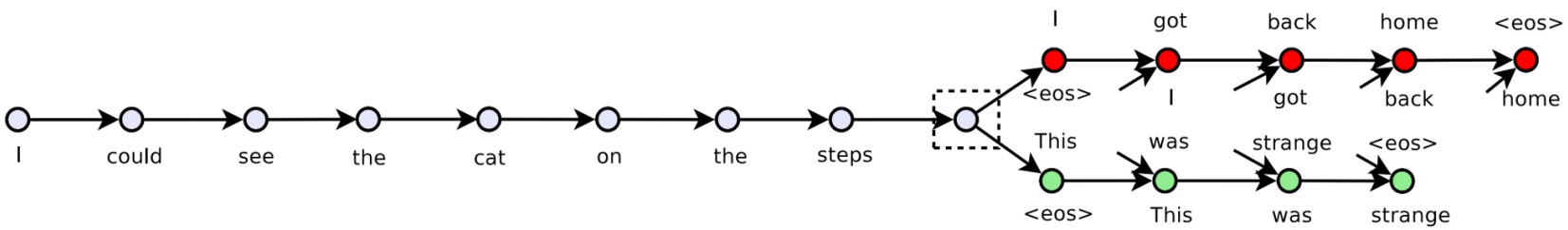


Figure taken from Kiros, Ryan, et al. "Skip-thought vectors" NIPS, 2015.

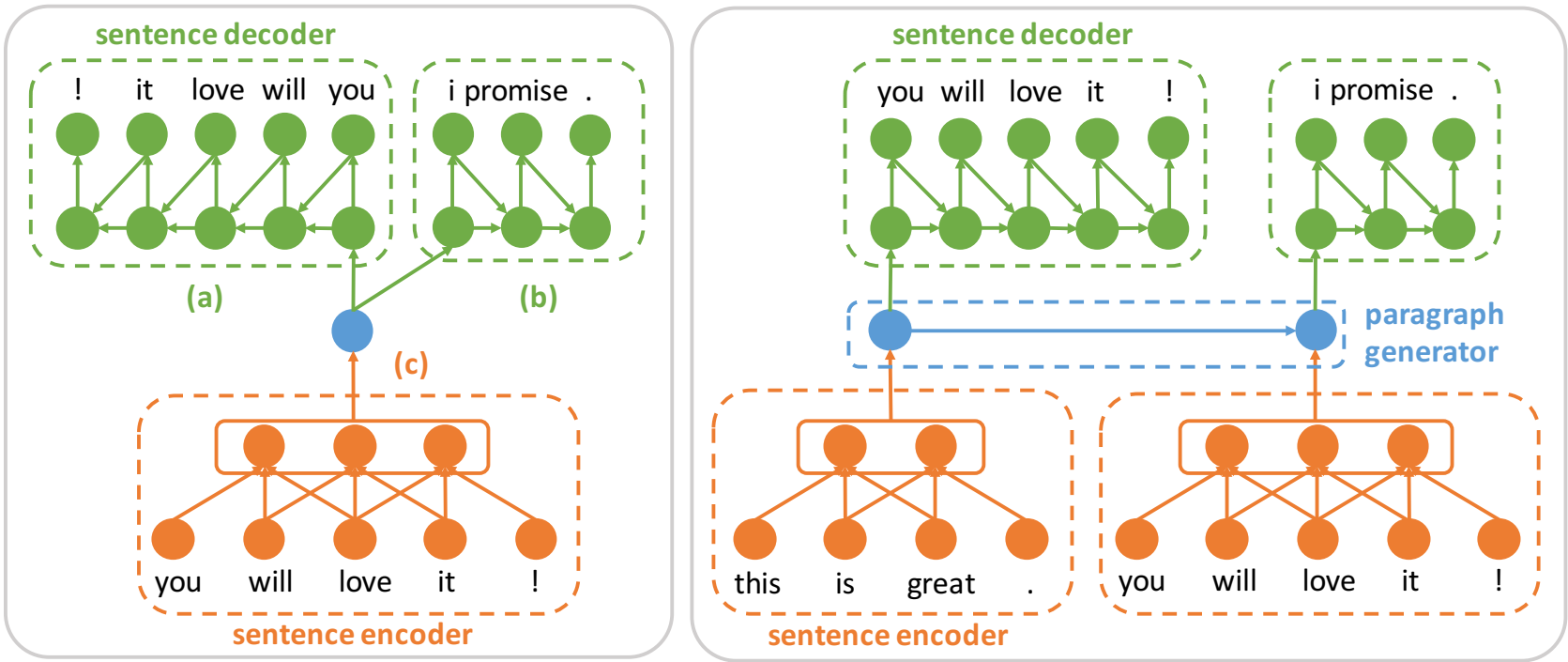
Contributions of this paper

- Model: CNN is used as the sentence encoder instead of RNN
  - CNN-LSTM model
  - hierarchical CNN-LSTM model
- Task: different tasks are considered, including
  - self-reconstruction
  - predicting multiple future sentences (a larger context window size is considered)
- Better empirical performance than skip-thought vectors

Outline

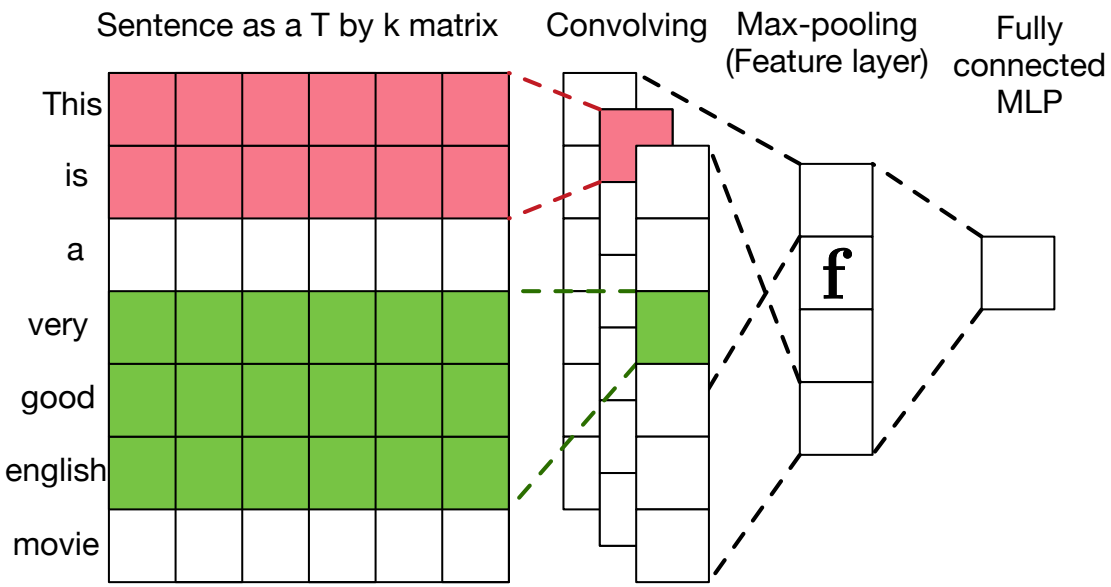
Model

- (Left) (a)+(c): autoencoder, capturing *intra*-sent. info.
- (Left) (b)+(c): future predictor, capturing *inter*-sent. info.
- (Left) (a)+(b)+(c): composite model, capturing both two
- (Right) hierarchical model, *longer-term* inter-sent. info.
  - Abstracting the RNN language model to the sentence level



CNN-LSTM model

- Use the CNN architecture in Kim (2014)[2]
- A sentence is represented as a matrix $\mathbf{X} \in \mathbb{R}^{k \times T}$, followed by a convolution operation.
- A max-over-time pooling operation is then applied.



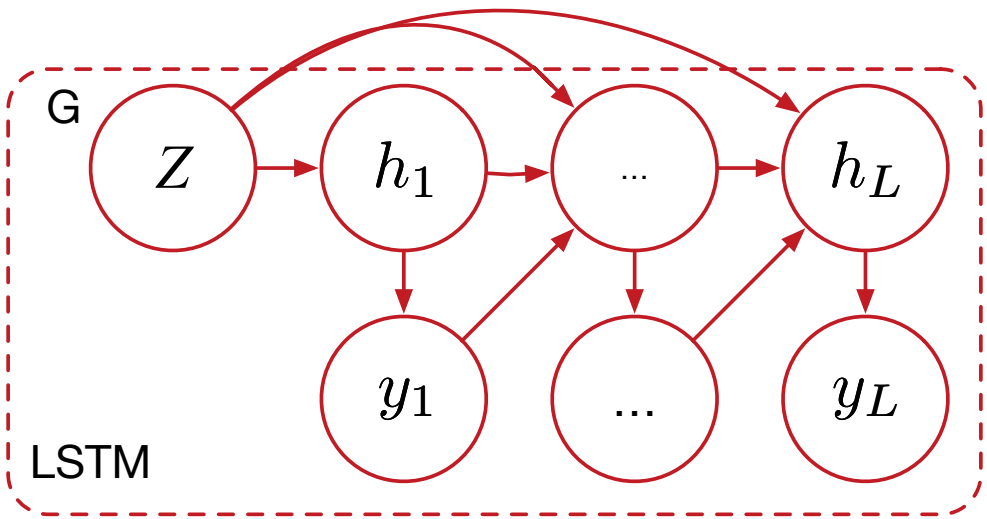
[2]Kim, Yoon. "Convolutional neural networks for sentence classification." EMNLP 2014.

CNN-LSTM model

- Many CNN variants: deeper, attention ...
- CNN *v.s.* LSTM: difficult to say which one is better.
- CNN typically requires fewer parameters due to the sparse connectivity, hence reducing memory requirements
  - our trained CNN encoder: 3M parameters;
  - skip-thought vector: 40M parameters
- CNN is easy to implement in parallel over the whole sentence, while LSTM needs sequential computation

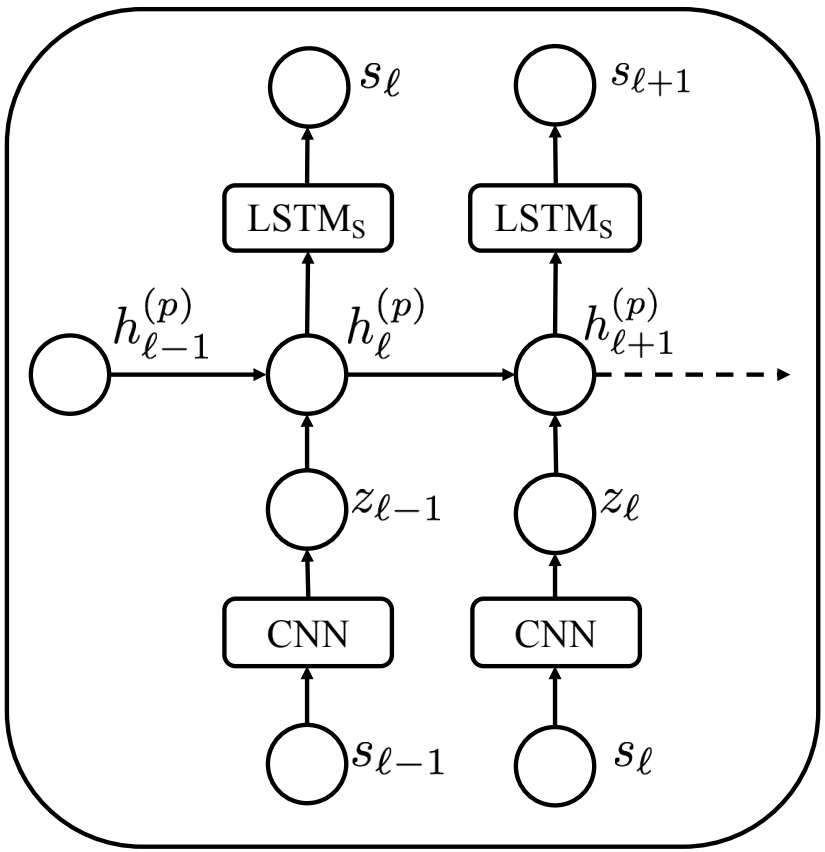
CNN-LSTM model

- LSTM decoder: translating latent code $\boldsymbol{z}$ into a sentence
- Objective: cross-entropy loss of predicting s_y given s_x

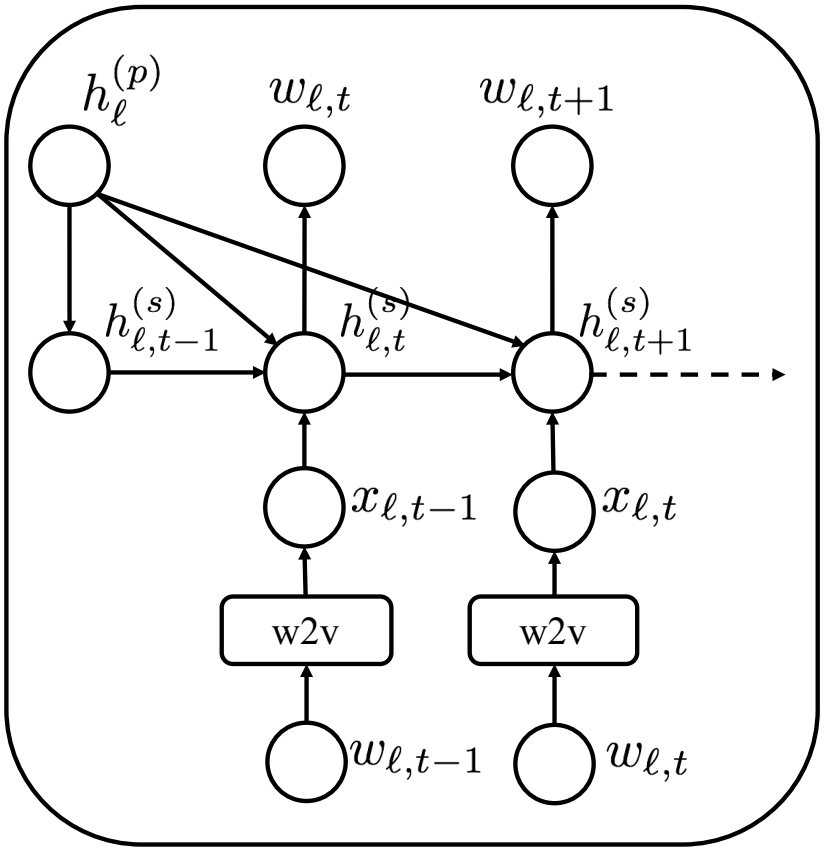


Hierarchical CNN-LSTM Model

This model characterizes the hierarchy *word-sentence-paragraph*.



(Left) LSTM$_P$



(Right) LSTM$_S$

Related work

- Learning generic sentence embedding
  - Skip-thought vector NIPS 2015
  - FastSent NAACL 2016
  - Towards universal paraphrastic sentence embeddings ICLR 2016
  - A simple but tough-to-beat baseline for sentence embeddings ICLR 2017
  - InferSent EMNLP 2017
  - ...
- CNN as encoder
  - image captioning
  - also utilzied for machine translation
- Hierarchical language modeling

Outline

Setup

- Tasks: 5 classification benchmarks, paraphrase detection, semantic relatedness and image-sentence ranking
- Training data: BookCorpus, 70M sentences over 7000 books
- CNN encoder: we employ filter windows of sizes $\{3,4,5\}$ with 800 feature maps each, hence 2400-dim.
- LSTM decoder: one hidden layer of 600 units.
- The CNN-LSTM models are trained with a vocabulary size of 22,154 words.
- Considering words not in the training set:
  - first we have pre-trained word embeddings $\mathcal{V}_{w2v}$
  - learn a linear transformation to map from $\mathcal{V}_{w2v}$ to $\mathcal{V}_{cnn}$
  - use fixed word embedding $\mathcal{V}_{w2v}$

Qualitative analysis - sentence retrieval

Query and nearest sentence
johnny nodded his curly head , and then his breath eased into an even rhythm . aiden looked at my face for a second , and then his eyes trailed to my extended hand .
i yelled in frustration , throwing my hands in the air . i stand up , holding my hands in the air .
i loved sydney , but i was feeling all sorts of homesickness . i loved timmy , but i thought i was a self-sufficient person .
“ i brought sad news to mistress betty , ” he said quickly , taking back his hand . “ i really appreciate you taking care of lilly for me , ” he said sincerely , handing me the money .
“ i am going to tell you a secret , ” she said quietly , and he leaned closer . “ you are very beautiful , ” he said , and he leaned in .
she kept glancing out the window at every sound , hoping it was jackson coming back . i kept checking the time every few minutes , hoping it would be five oclock .
leaning forward , he rested his elbows on his knees and let his hands dangle between his legs . stepping forward , i slid my arms around his neck and then pressed my body flush against his .
i take tris ’s hand and lead her to the other side of the car , so we can watch the city disappear behind us . i take emma ’s hand and lead her to the first taxi , everyone else taking the two remaining cars .

Qualitative analysis - vector "compositionality"

- word vector compositionality[3]
  - *king - man + woman = queen*
- sentence vector compositionality
  - We calculate $\boldsymbol{z}^{\star}=\boldsymbol{z}(\mathrm{A})-\boldsymbol{z}(\mathrm{B})+\boldsymbol{z}(\mathrm{C})$, which is sent to the LSTM to generate sentence D.

A	you needed me?	this is great.	its lovely to see you.	he had thought he was going crazy.
B	you got me?	this is awesome.	its great to meet you.	i felt like i was going crazy.
C	i got you.	you are awesome.	its great to meet him.	i felt like to say the right thing.
D	i needed you.	you are great.	its lovely to see him.	he had thought to say the right thing.

[3] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." NIPS 2013.

Quantitative results - classification & paraphrase detection

- composite model > autoencoder > future predictor
- hierarchical model > future predictor
- combine > composite model > hierarchical model

Method	**MR**	**CR**	**SUBJ**	**MPQA**	**TREC**	**MSRP(Acc/F1)**
Our Results						
autoencoder	75.53	78.97	91.97	87.96	89.8	73.61 / 82.14
future predictor	72.56	78.44	90.72	87.48	86.6	71.87 / 81.68
hierarchical model	75.20	77.99	91.66	88.21	90.0	73.96 / 82.54
composite model	76.34	79.93	92.45	88.77	91.4	74.65 / 82.21
combine	77.21	80.85	93.11	89.09	91.8	75.52 / 82.62

Quantitative results - classification & paraphrase detection

- Using (fixed) pre-trained word embeddings consistently provides better performance than using the learned word embeddings.

Method	**MR**	**CR**	**SUBJ**	**MPQA**	**TREC**	**MSRP(Acc/F1)**
Our Results						
hierarchical model	75.20	77.99	91.66	88.21	90.0	73.96 / 82.54
composite model	76.34	79.93	92.45	88.77	91.4	74.65 / 82.21
combine	77.21	80.85	93.11	89.09	91.8	75.52 / 82.62
hierarchical model+emb.	75.30	79.37	91.94	88.48	90.4	74.25 / 82.70
composite model+emb.	77.16	80.64	92.14	88.67	91.2	74.88 / 82.28
combine+emb.	**77.77**	**82.05**	**93.63**	**89.36**	**92.6**	**76.45 / 83.76**

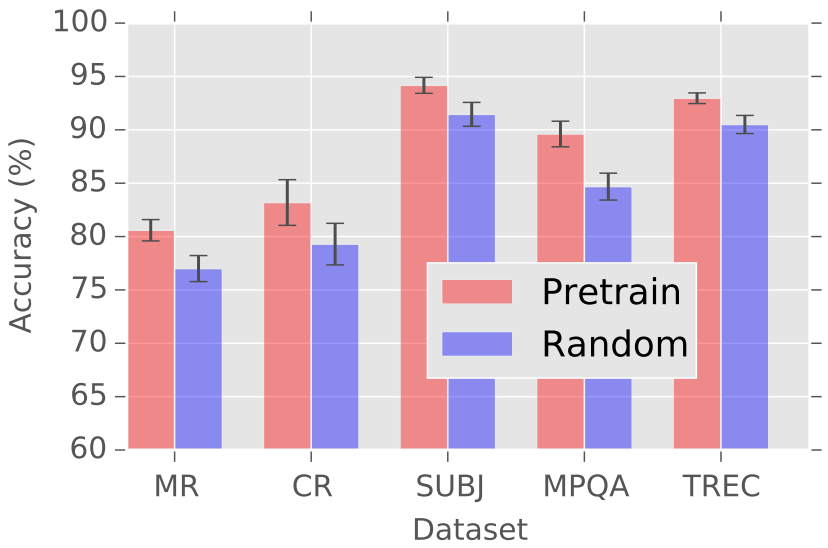
Quantitative results - classification & paraphrase detection

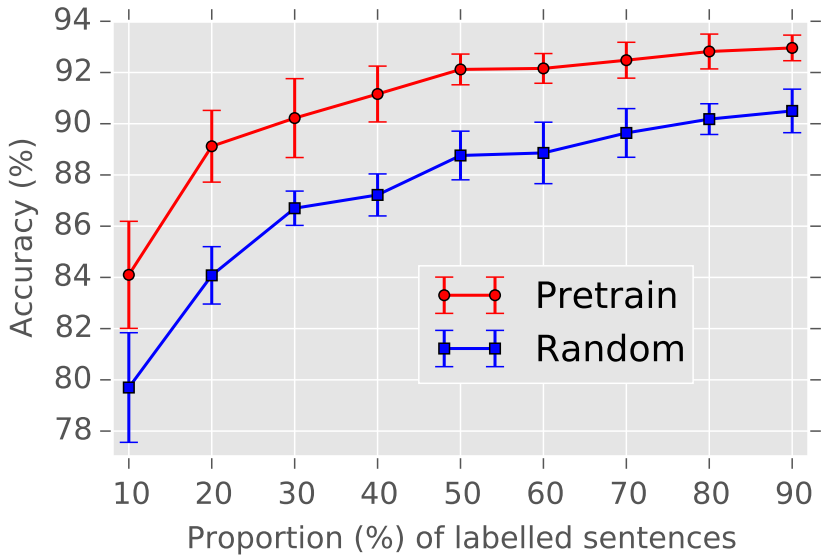
- Our model provides better results than skip-thought vectors.
- Generic methods performs worse than task-dependent methods.

Method	**MR**	**CR**	**SUBJ**	**MPQA**	**TREC**	**MSRP(Acc/F1)**
Generic						
SDAE+emb.	74.6	78.0	90.8	86.9	78.4	73.7 / 80.7
FastSent	70.8	78.4	88.7	80.6	76.8	72.2 / 80.3
skip-thought	76.5	80.1	93.6	87.1	92.2	73.0 / 82.0
Ours	**77.77**	**82.05**	**93.63**	**89.36**	**92.6**	**76.45** / **83.76**
Task-dependent						
CNN	81.5	85.0	93.4	89.6	93.6	–
AdaSent	83.1	86.3	95.5	93.3	92.4	–
Bi-CNN-MI	–	–	–	–	–	78.1/84.4
MPSSM-CNN	–	–	–	–	–	78.6/84.7

Quantitative results - classification & paraphrase detection

- Pretraining means initializing the CNN parameters using the learned generic encoder.
- The pretraining provides substantial improvements over random initialization.
- As the size of the set of labeled sentences grows, the improvement becomes smaller, as expected.





Quantitative results - semantic relatedness

- Similiar observation also holds true for semantic relatedness and image-sentence retrieval tasks.

Method	r	ρ	**MSE**
skip-thought	0.8584	0.7916	0.2687
Our Results			
hierarchical model	0.8333	0.7646	0.3135
composite model	0.8434	0.7767	0.2972
combine	0.8533	0.7891	0.2791
hierarchical model+emb.	0.8352	0.7588	0.3152
composite model+emb.	0.8500	0.7867	0.2872
combine+emb.	**0.8618**	**0.7983**	**0.2668**
Task-dependent methods			
Tree-LSTM	0.8676	0.8083	0.2532

Quantitative results - image-sentence retrieval

- Similiar observation also holds true for semantic relatedness and image-sentence retrieval tasks.

	Image Annotation		Image Search	
Method	**R@1**	**Med *r***	**R@1**	**Med *r***
uni-skip	30.6	3	22.7	4
bi-skip	32.7	3	24.2	4
combine-skip	33.8	3	25.9	4
Our Results				
hierarchical model+emb.	32.7	3	25.3	4
composite model+emb.	33.8	3	25.7	4
combine+emb.	**34.4**	3	**26.6**	4
Task-dependent methods				
DVSA	38.4	1	27.4	3
m-RNN	41.0	2	29.0	3

Outline

Take away

- Conclusion in Skip-Thought paper

> We believe our model for learning skip-thought vectors only scratches the surface of possible objectives. Many variations have yet to be explored, including (a) deep encoders and decoders, (b) larger context windows, (c) encoding and decoding paragraphs, (d) other encoders, such as convnets. It is likely the case that more exploration of this space will result in even higher quality representations.

- Inspired by skip-thought, we considered
  - different encoders, such as CNN; save parameters, more parallelizable
  - different tasks, including reconstruction and use of larger context windows
- and achieved promising performance

Follow-up work

- Q: How to learn a better sentence/paragraph representation?
- A: Deconvolutional Paragraph Representation Learning
  NIPS 2017
  - deeper CNN encoder
  - *fully deconvolutional decoder*
  - tries to slove the teacher forcing and exposure bias problems
  - used for (semi-)supervised learning

Thank You