Paper sharing

——ICLR2020

- REDUCING TRANSFORMER DEPTH ON DEMAND WITH STRUCTURED DROPOUT
- ON THE RELATIONSHIP BETWEEN SELF-ATTENTION AND CONVOLUTIONAL LAYERS
- FEW-SHOT TEXT CLASSIFICATION WITH DISTRIBUTIONAL SIGNATURES

2020/5/27 ZhuJingdan

REDUCING TRANSFORMER DEPTH ON DEMAND WITH STRUCTURED DROPOUT

Angela Fan
Facebook AI Research/LORIA
angelafan@fb.com

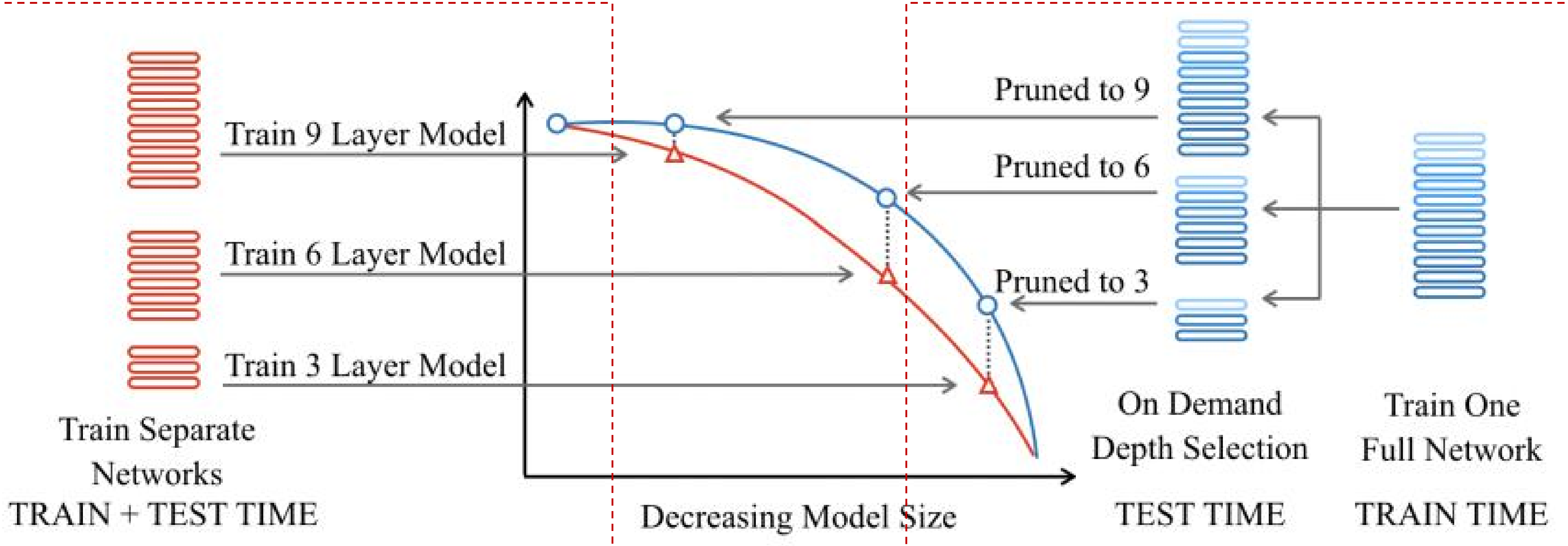
Edouard Grave
Facebook AI Research
egrave@fb.com

Armand Joulin
Facebook AI Research
ajoulin@fb.com

∵Overparameterized transformer networks contain hundreds of millions of parameters, necessitating a large amount of computation and making them prone to overfitting.

∴They explore LayerDrop, a form of structured dropout, which has a regularization effect during training and allows for efficient pruning at inference time.

It is possible to select sub-networks ofany depth from one large network without having to finetune them and with limited impact on performance.



In contrast to standard approaches that must re-train a new model from scratch for each model size , **this method trains only one network from which multiple shallow models can be extracted**.

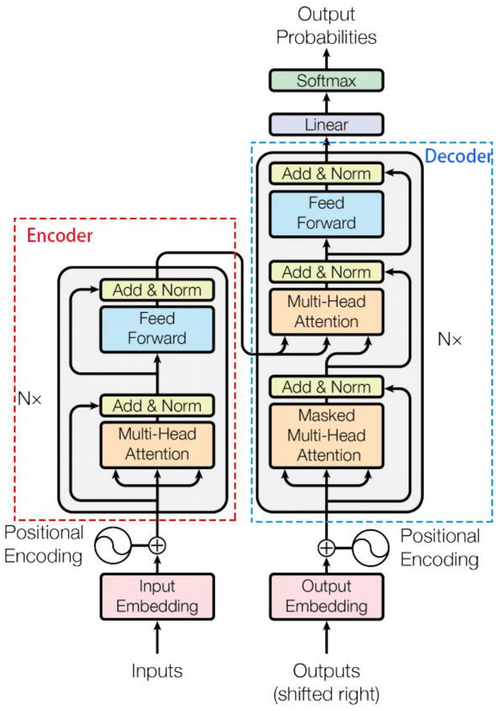
LayerDrop **randomly drops layers** at training time.

At test time, this allows for sub-network selection to any desired depth as the network has been trained to **be robust to pruning**.

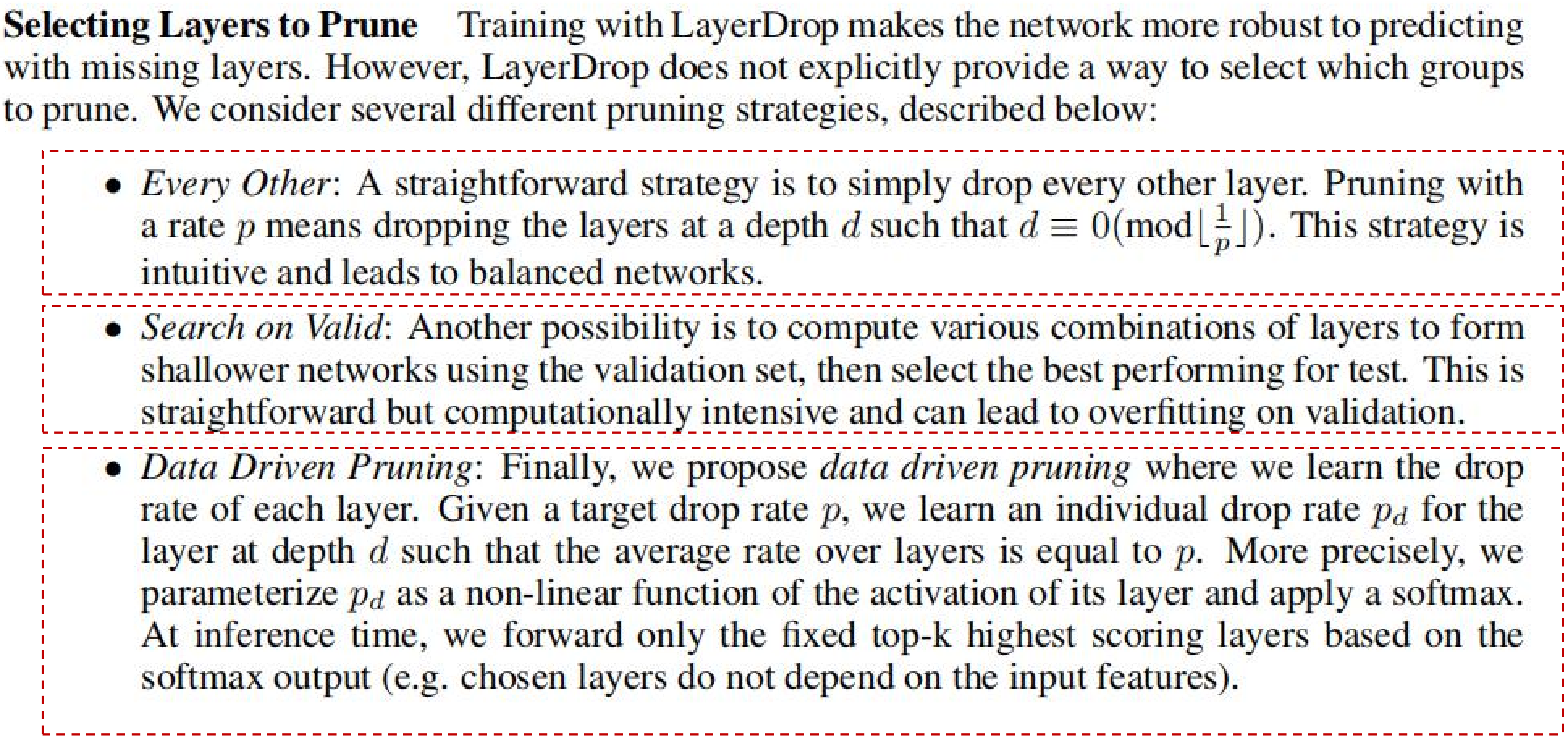
$$\mathbf{Y} = \mathrm{Softmax}(\mathbf{X}^T\mathbf{K}(\mathbf{QX}+\mathbf{P}))\mathbf{VX}$$

$$\mathtt{FFN}(\mathbf{x}) = \mathbf{U}\ \mathtt{ReLU}\left(\mathbf{Vx}\right)$$

$$\forall i,\ \mathbf{M}[i] \in \{0,1\}, \quad \text{and } \forall G \in \mathcal{G},\ \forall (i,j) \in G,\ \mathbf{M}[i] = \mathbf{M}[j].$$



Dropping weights using groups that follow some of these inherent structures potentially leads to a significant reduction of the inference time. This is equivalent to constraining the mask M to be constant over some predefined groups of weights. More precisely, given a set G of predefined groups of weights, the {0, 1} mask matrix M is randomly sampled over groups instead of weights:

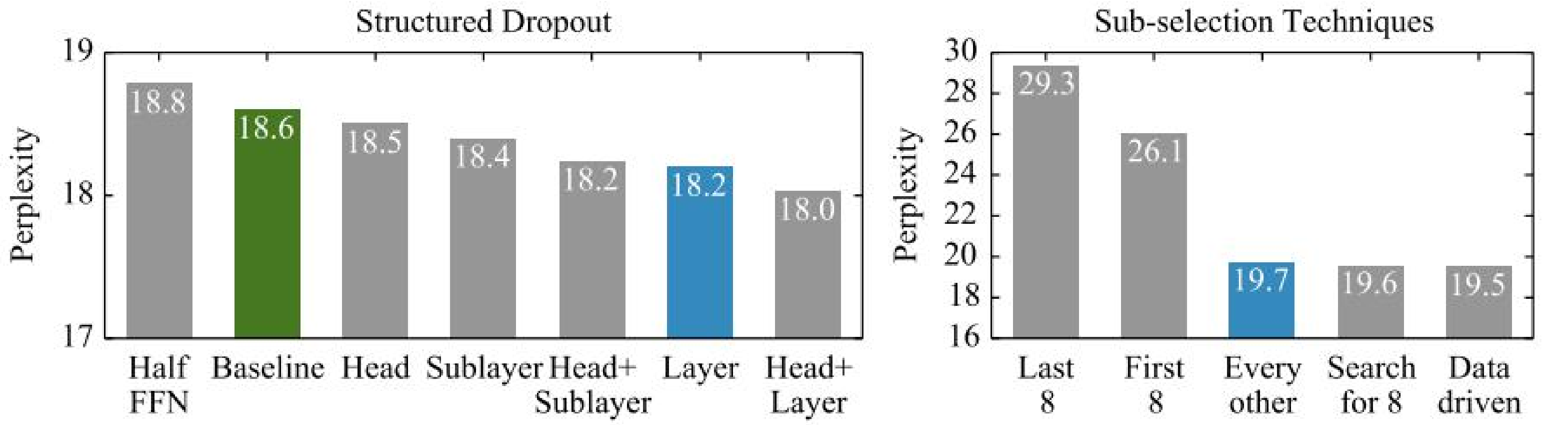


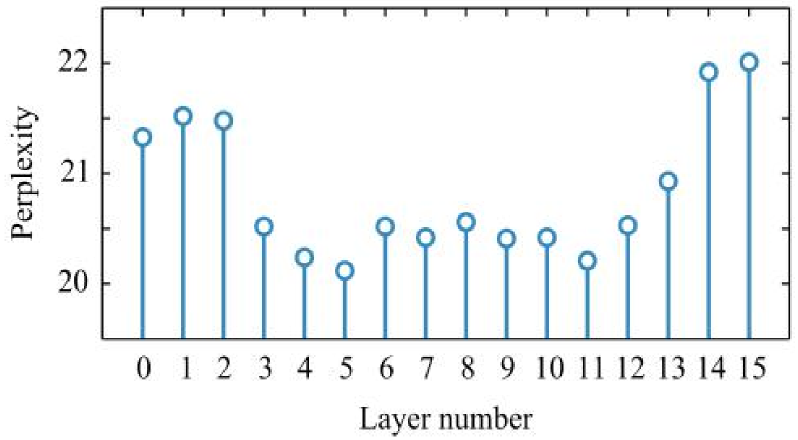
Selecting Layers to Prune Training with LayerDrop makes the network more robust to predicting with missing layers. However, LayerDrop does not explicitly provide a way to select which groups to prune. We consider several different pruning strategies, described below:

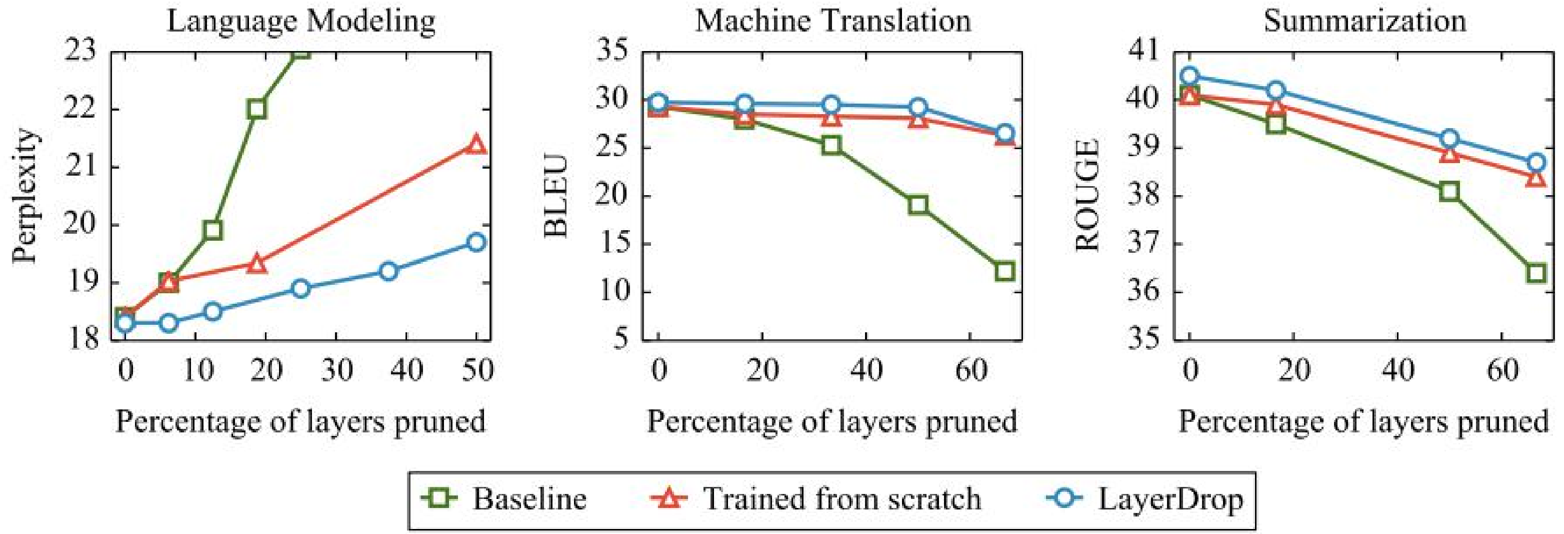
- *Every Other*: A straightforward strategy is to simply drop every other layer. Pruning with a rate p means dropping the layers at a depth d such that $d \equiv 0(\text{mod}\lfloor \frac{1}{p} \rfloor)$. This strategy is intuitive and leads to balanced networks.

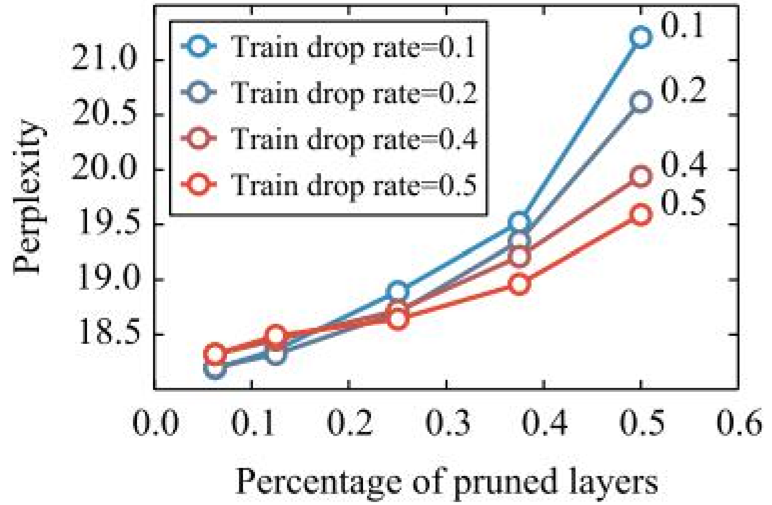
- *Search on Valid*: Another possibility is to compute various combinations of layers to form shallower networks using the validation set, then select the best performing for test. This is straightforward but computationally intensive and can lead to overfitting on validation.

- *Data Driven Pruning*: Finally, we propose *data driven pruning* where we learn the drop rate of each layer. Given a target drop rate p, we learn an individual drop rate p_d for the layer at depth d such that the average rate over layers is equal to p. More precisely, we parameterize p_d as a non-linear function of the activation of its layer and apply a softmax. At inference time, we forward only the fixed top-k highest scoring layers based on the softmax output (e.g. chosen layers do not depend on the input features).

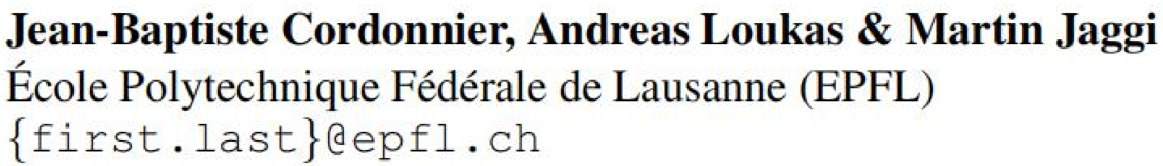

Structured Dropout
Perplexity
19
18
17
18.8
18.6
18.5
18.4
18.2
18.2
18.0
Half FFN
Baseline
Head
Sublayer
Head+ Sublayer
Layer
Head+ Layer
Sub-selection Techniques
Perplexity
30
28
26
24
22
20
18
16
29.3
26.1
19.7
19.6
19.5
Last 8
First 8
Every other
Search for 8
Data driven


Perplexity
22
21
20
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Layer number


Language Modeling
Perplexity
23
22
21
20
19
18
0
10
20
30
40
50
Percentage of layers pruned
Machine Translation
BLEU
35
30
25
20
15
10
5
0
20
40
60
Percentage of layers pruned
Summarization
ROUGE
41
40
39
38
37
36
35
0
20
40
60
Percentage of layers pruned
Baseline
Trained from scratch
LayerDrop


Train drop rate=0.1
Train drop rate=0.2
Train drop rate=0.4
Train drop rate=0.5
0.1
0.2
0.4
0.5
Perplexity
21.0
20.5
20.0
19.5
19.0
18.5
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Percentage of pruned layers

ON THE RELATIONSHIP BETWEEN SELF-ATTENTION AND CONVOLUTIONAL LAYERS



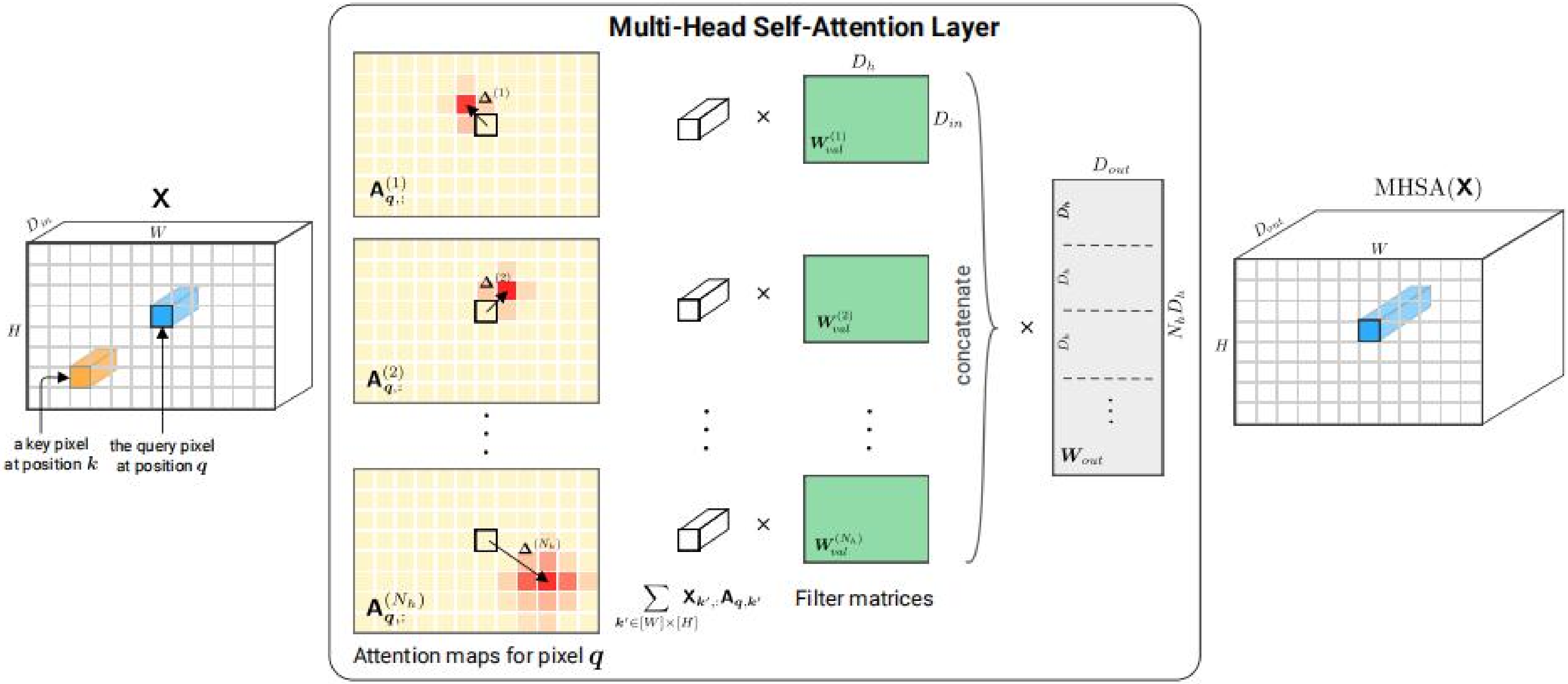
Jean-Baptiste Cordonnier, Andreas Loukas & Martin Jaggi
École Polytechnique Fédérale de Lausanne (EPFL)
{first.last}@epfl.ch

- From a theoretical perspective, they provide a constructive proof showing that self-attention layers can express any convolutional layers.
- Their experiments show that the first few layers of attention-only architectures (Ramachandran et al., 2019) do learn to attend on grid-like pattern around each query pixel.

Code: github.com/epfml/attention-cnn.
Website: epfml.github.io/attention-cnn.

This section derives sufficient conditions such that a multi-head self-attention layer can simulate a convolutional layer. Our main result is the following:

Theorem 1. *A multi-head self-attention layer with N_h heads of dimension D_h, output dimension D_{out} and a relative positional encoding of dimension $D_p \geq 3$ can express any convolutional layer of kernel size $\sqrt{N_h} \times \sqrt{N_h}$ and $\min(D_h, D_{out})$ output channels.*



- *Paddong*
- *Stride*
- *Dilation*

scores of each self-attention head should attend to a different relative shift within the set $\Delta_K = \{-\lfloor K/2 \rfloor, \dots, \lfloor K/2 \rfloor\}^2$ of all pixel shifts in a $K \times K$ kernel. The exact condition can be found in

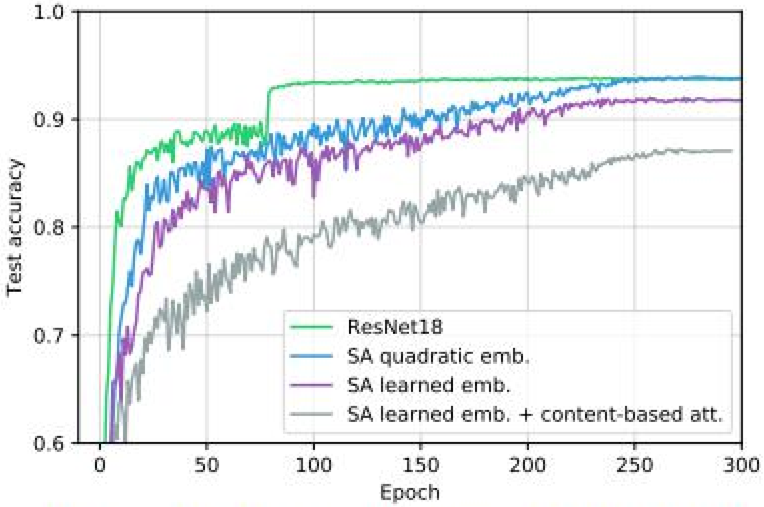
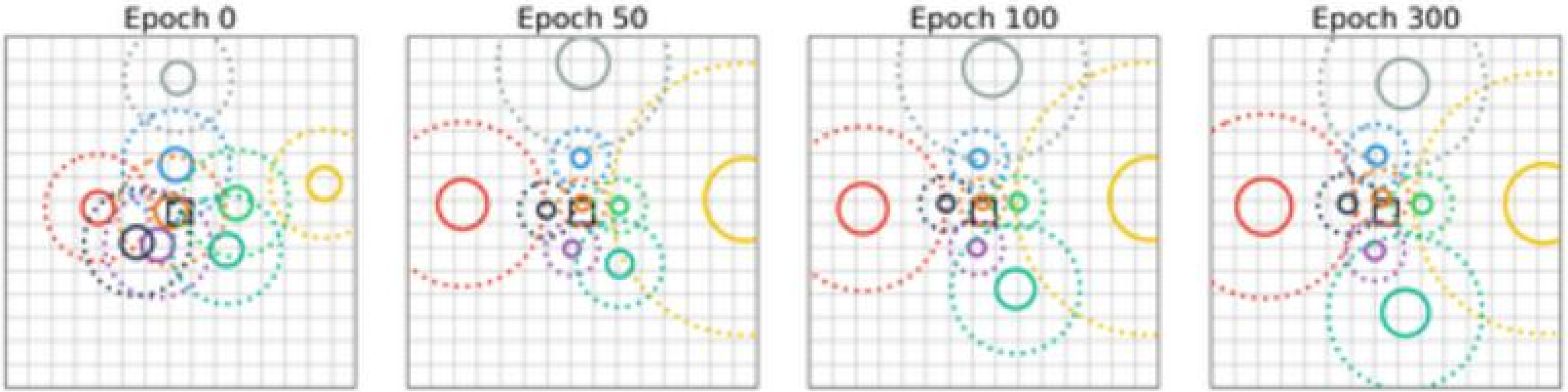


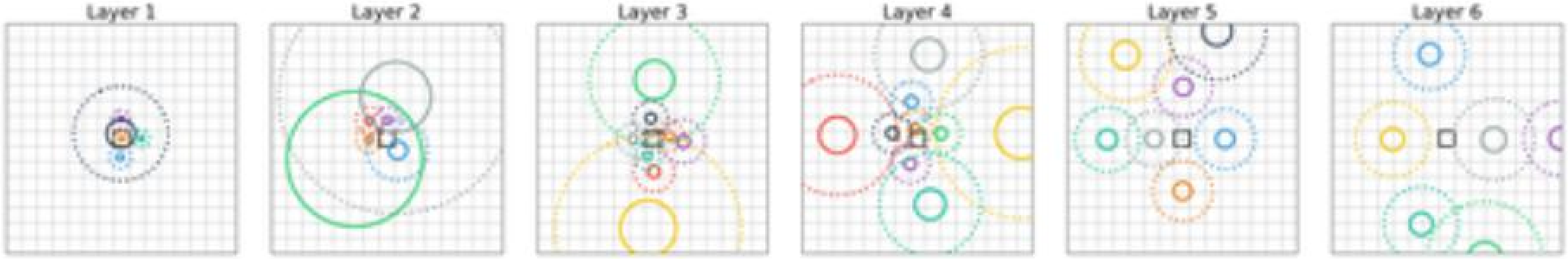
Figure 2: Test accuracy on CIFAR-10.

$$\boldsymbol{v}^{(h)} := -\alpha^{(h)}\,(1, -2\boldsymbol{\Delta}_1^{(h)}, -2\boldsymbol{\Delta}_2^{(h)}) \quad \boldsymbol{r}_{\boldsymbol{\delta}} := (\|\boldsymbol{\delta}\|^2, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2) \quad \boldsymbol{W}_{qry} = \boldsymbol{W}_{key} := \boldsymbol{0} \quad \widehat{\boldsymbol{W}_{key}} := \boldsymbol{I} \quad (9)$$

Models	accuracy	# of params	# of FLOPS
ResNet18	0.938	11.2M	1.1B
SA quadratic emb.	0.938	12.1M	6.2B
SA learned emb.	0.918	12.3M	6.2B
SA learned emb. + content	0.871	29.5M	15B

Table 1: Test accuracy on CIFAR-10 and model sizes. SA stands for Self-Attention.





Paper sharing

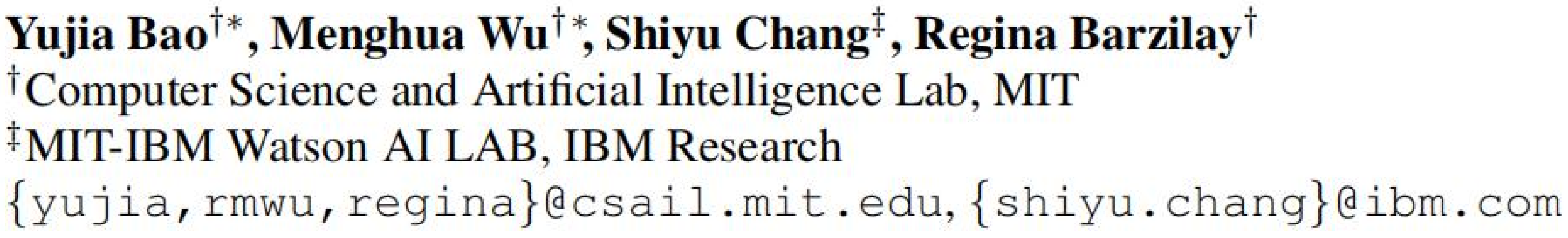
——接上周

- FEW-SHOT TEXT CLASSIFICATION WITH DISTRIBUTIONAL SIGNATURES

- FEW-SHOT NATURAL LANGUAGE GENERATION FOR TASK-ORIENTED DIALOG

2020/6/3 ZhuJingdan

FEW-SHOT TEXT CLASSIFICATION WITH DISTRIBUTIONAL SIGNATURES



Yujia Bao[†*], Menghua Wu[†*], Shiyu Chang[‡], Regina Barzilay[†]
[†]Computer Science and Artificial Intelligence Lab, MIT
[‡]MIT-IBM Watson AI LAB, IBM Research
{yujia,rmwu,regina}@csail.mit.edu, {shiyu.chang}@ibm.com

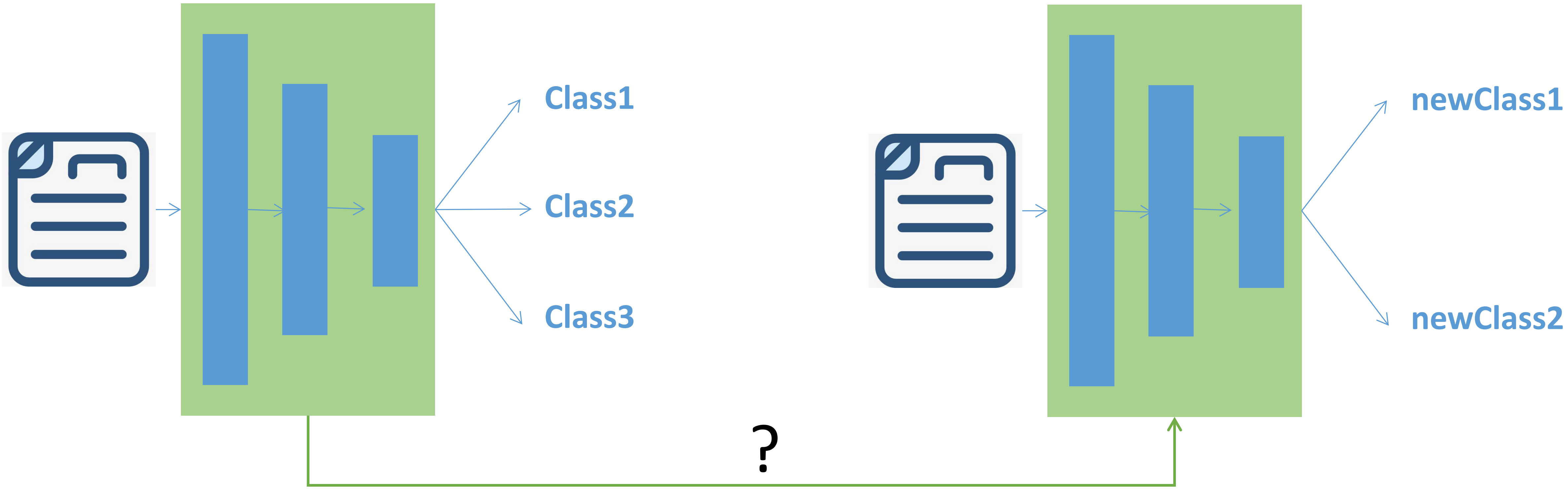
They explore meta-learning for few-shot text classification.

∵Directly applying meta-learning to text is challenging–lexical features highly informative for one task may be insignificant for another.
∴Rather than learning solely from words, their model also leverages their distributional signatures, which encode pertinent wordoccurrence patterns.

Problem

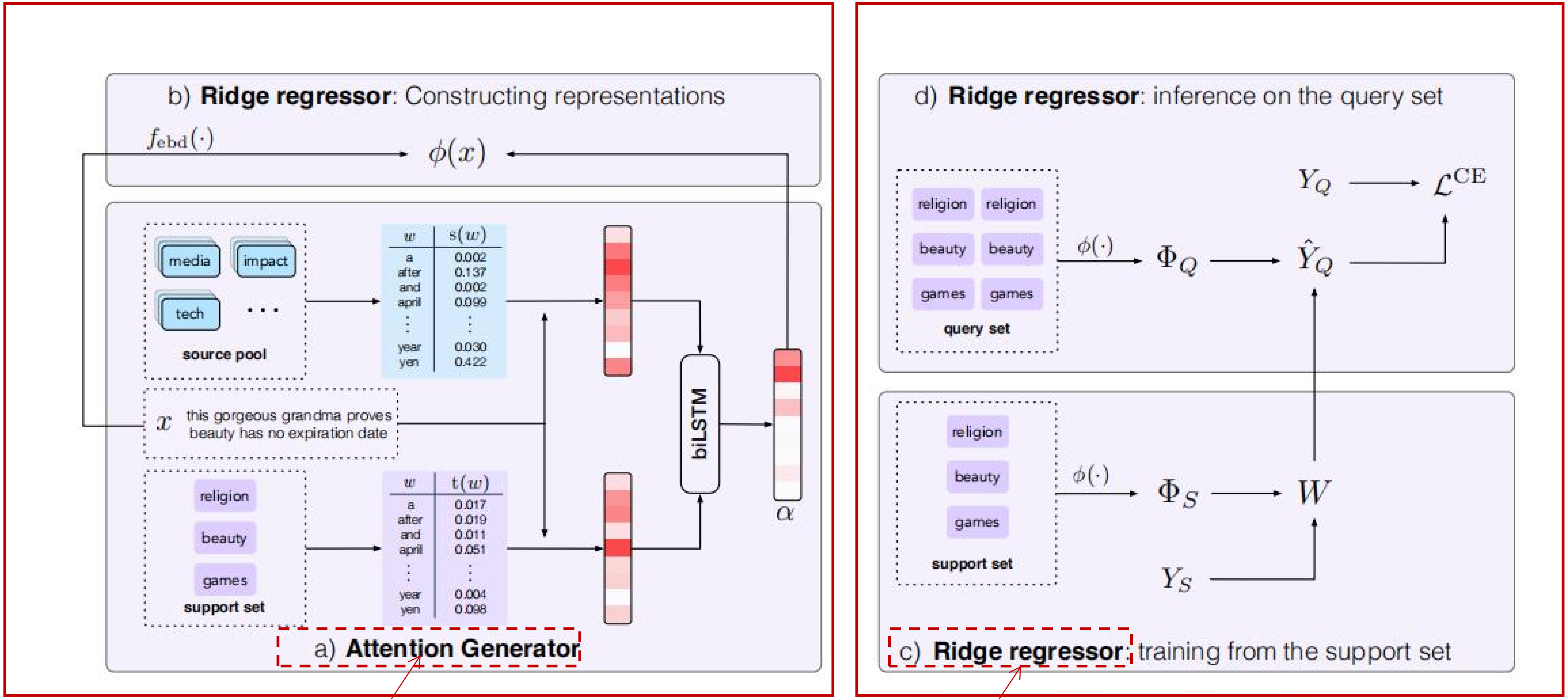
new classes

only a few instances,.g, only one



The problem is how to train a model on a dataset that have many instances for each class and how to **adapt this model to a new dataset which only has a few instances** for each class.

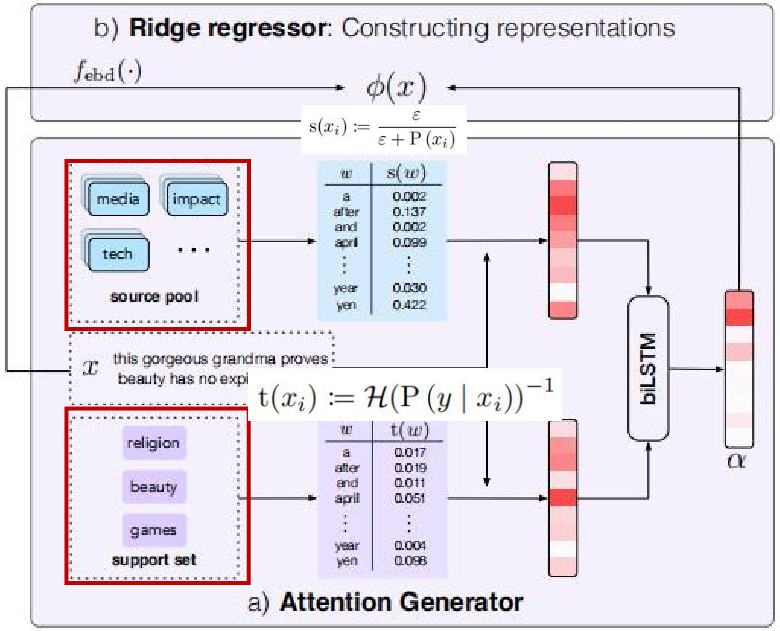
(The classes in the new dataset are unseen in the first one.)



w	$s(w)$
a	0.002
after	0.137
and	0.002
april	0.099
⋮	⋮
year	0.030
yen	0.422

w	$t(w)$
a	0.017
after	0.019
and	0.011
april	0.051
⋮	⋮
year	0.004
yen	0.098

Figure 5: Illustration of our model for an episode with $N = 3, K = 1, L = 2$. The attention generator translates the distributional signatures from the source pool and the support set into an attention α for each input example x (5a). The ridge regressor utilizes the generated attention to weight the lexical representations (5b). It then learns from the support set (5c) and makes predictions over the query set (5d).

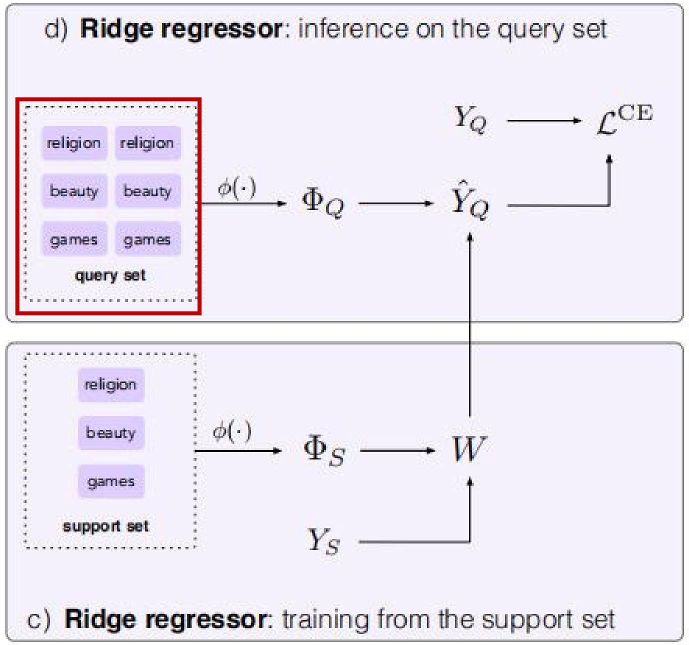

b) Ridge regressor: Constructing representations
$f_{\text{ebd}}(\cdot)$
$\phi(x)$
$\mathrm{s}(x_i) := \frac{\varepsilon}{\varepsilon + \mathrm{P}(x_i)}$
media
impact
tech
...
source pool
x
this gorgeous grandma proves beauty has no expi
$\mathrm{t}(x_i) := \mathcal{H}(\mathrm{P}(y \mid x_i))^{-1}$
religion
beauty
games
support set
biLSTM
α
a) Attention Generator

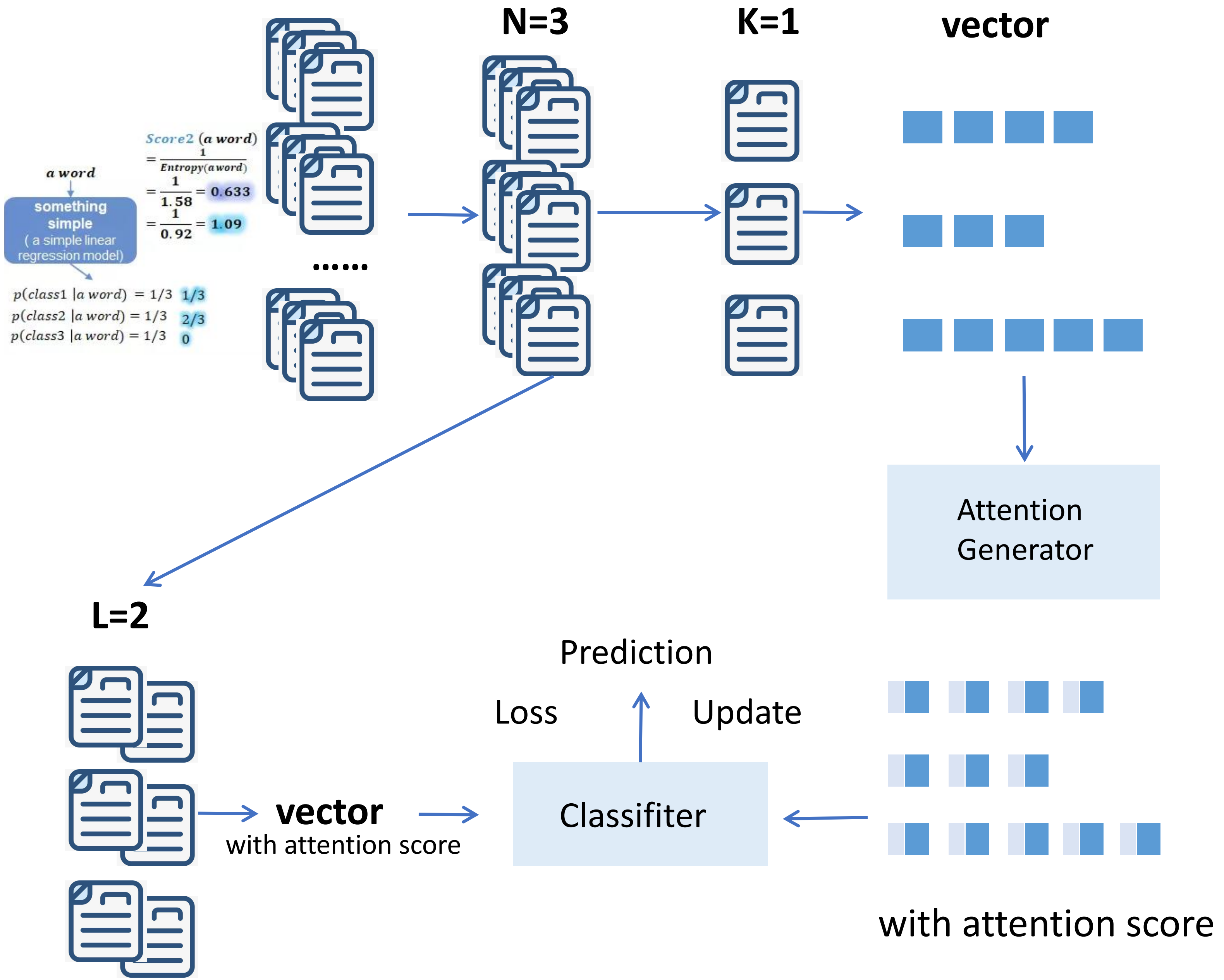
w	$\mathrm{s}(w)$
a	0.002
after	0.137
and	0.002
april	0.099
⋮	⋮
year	0.030
yen	0.422

w	$\mathrm{t}(w)$
a	0.017
after	0.019
and	0.011
april	0.051
⋮	⋮
year	0.004
yen	0.098

ode with $N = 3,\ K = 1,\ L = 2.$

from the source pool and the supp





$$Score2\ (a\ word) = \frac{1}{Entropy(a\,word)} = \frac{1}{1.58} = 0.633 = \frac{1}{0.92} = 1.09$$

$p(class1\ |a\ word) = 1/3$ 1/3

$p(class2\ |a\ word) = 1/3$ 2/3

$p(class3\ |a\ word) = 1/3$ 0

5.1 DATASETS

We evaluate our approach on five text classification datasets and one relation classification dataset.[5] (See Appendix A.4 for more details.)

20 Newsgroups is comprised of informal discourse from news discussion forums (Lang, 1995). Documents are organized under 20 topics.

RCV1 is a collection of Reuters newswire articles from 1996 to 1997 (Lewis et al., 2004). These articles are written in formal speech and labeled with a set of topic codes. We consider 71 second-level topics as our total class set and discard articles that belong to more than one class.

Reuters-21578 is a collection of shorter Reuters articles from 1987 (Lewis, 1997). We use the standard ApteMod version of the dataset. We discard articles with more than one label and consider 31 classes that have at least 20 articles.

Amazon product data contains customer reviews from 24 product categories (He & McAuley, 2016). Our goal is to classify reviews into their respective product categories. Since the original dataset is notoriously large (142.8 million reviews), we select a more tractable subset by sampling 1000 reviews from each category.

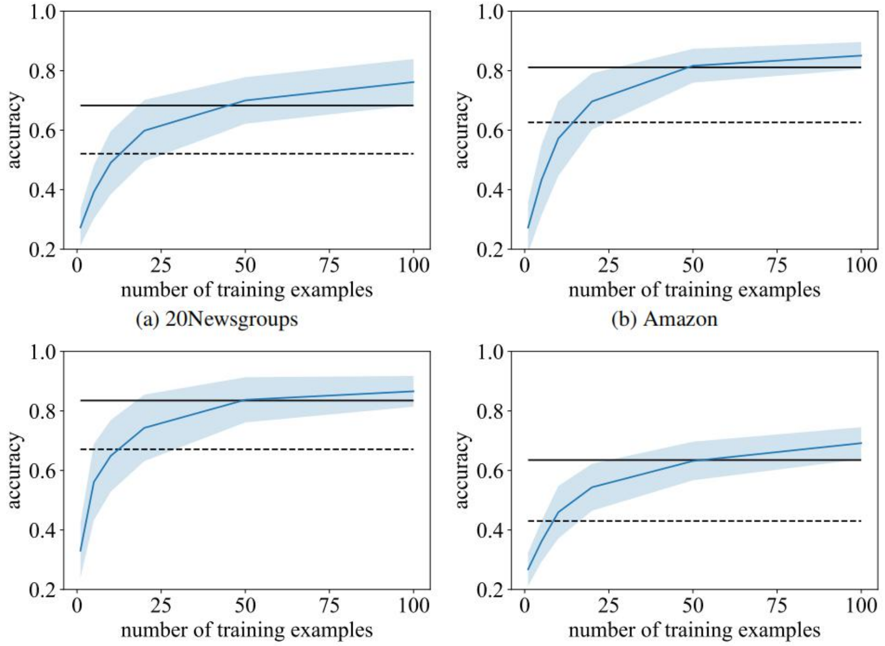
HuffPost headlines consists of news headlines published on HuffPost between 2012 and 2018 (Misra, 2018). These headlines split among 41 classes. They are shorter and less grammatical than formal sentences.

FewRel is a relation classification dataset developed for few-shot learning (Han et al., 2018). Each example is a single sentence, annotated with a head entity, a tail entity, and their relation. The goal is to predict the correct relation between the head and tail. The public dataset contains 80 relation types.

Method		20 News		Amazon		HuffPost		RCV1		Reuters		FewRel		Average	
Rep.	Alg.	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot	1 shot	5 shot
AVG	NN	33.9	45.8	46.7	60.3	31.4	41.5	43.7	60.8	56.5	80.5	47.5	60.6	43.3	58.2
IDF	NN	38.8	51.9	51.4	67.1	31.5	42.3	41.9	58.2	57.8	82.9	46.8	60.6	44.7	60.5
CNN	FT	33.0	47.1	45.7	63.9	32.4	44.1	40.3	62.3	70.9	91.0	54.0	71.1	46.0	63.2
AVG	PROTO	36.2	45.4	37.2	51.9	35.6	41.6	28.4	31.2	59.5	68.1	44.0	46.5	40.1	47.4
IDF	PROTO	37.8	46.5	41.9	59.2	34.8	50.2	32.1	35.6	61.0	72.1	43.0	61.9	41.8	54.2
CNN	PROTO	29.6	35.0	34.0	44.4	33.4	44.2	28.4	29.3	65.2	74.3	49.7	65.1	40.1	48.7
AVG	MAML	33.7	43.9	39.3	47.2	36.1	49.6	39.9	50.6	54.6	62.5	43.8	57.8	41.2	51.9
IDF	MAML	37.2	48.6	43.6	62.4	38.9	53.7	42.5	54.1	61.5	72.0	48.2	65.8	45.3	59.4
CNN	MAML	28.9	36.7	35.3	43.7	34.1	45.8	39.0	51.1	66.6	85.0	51.7	66.9	42.6	54.9
AVG	RR	37.6	57.2	50.2	72.7	36.3	54.8	48.1	72.6	63.4	90.0	53.2	72.2	48.1	69.9
IDF	RR	44.8	64.3	60.2	79.7	37.6	59.5	48.6	72.8	69.1	93.0	55.6	75.3	52.6	74.1
CNN	RR	32.2	44.3	37.3	53.8	37.3	49.9	41.8	59.4	71.4	87.9	56.8	71.8	46.1	61.2
OUR		**52.1**	**68.3**	**62.6**	**81.1**	**43.0**	**63.5**	**54.1**	**75.3**	**81.8**	**96.0**	**67.1**	**83.5**	**60.1**	**78.0**
OUR w/o t(·)		50.1	67.5	61.7	80.5	42.0	60.8	51.5	75.1	76.7	93.7	66.9	83.2	58.1	76.8
OUR w/o s(·)		41.9	60.7	51.1	75.3	40.1	60.2	48.5	72.8	78.1	94.8	65.8	82.6	54.2	74.4
OUR w/o biLSTM		50.3	66.9	61.9	80.9	42.2	63.0	51.8	74.1	77.2	95.4	66.4	82.9	58.3	77.2
OUR w EBD		39.7	57.5	56.5	76.3	40.6	58.6	48.6	71.5	81.7	95.8	61.5	80.9	54.8	73.4

Table 1: Results of 5-way 1-shot and 5-way 5-shot classification on six datasets. The bottom four rows present our ablation study. For complete results with standard deviations see Table 8 and 9 in Appendix A.12.

class	input example
taste	you wo n't even miss the meat with these delicious vegetarian sandwiches
taste	these cookies are spot - on copies of the oscars dresses
word news	prime minister saad hariri 's return to lebanon : a moment of truth
word news	new zealand just became the 11th country to send a rocket into orbit
style	beyoncé dressed like the queen she is at the grammys
style	tilda swinton , is that a jacket or a dress ?
science	the world of science has a lot to look forward to in 2016
science	dione crosses saturn 's disk in spectacular new image
education	the global search for education : just imagine secretary hargreaves
education	thinking at harvard : what is the future of learning ?



(a) 20Newsgroups

(b) Amazon

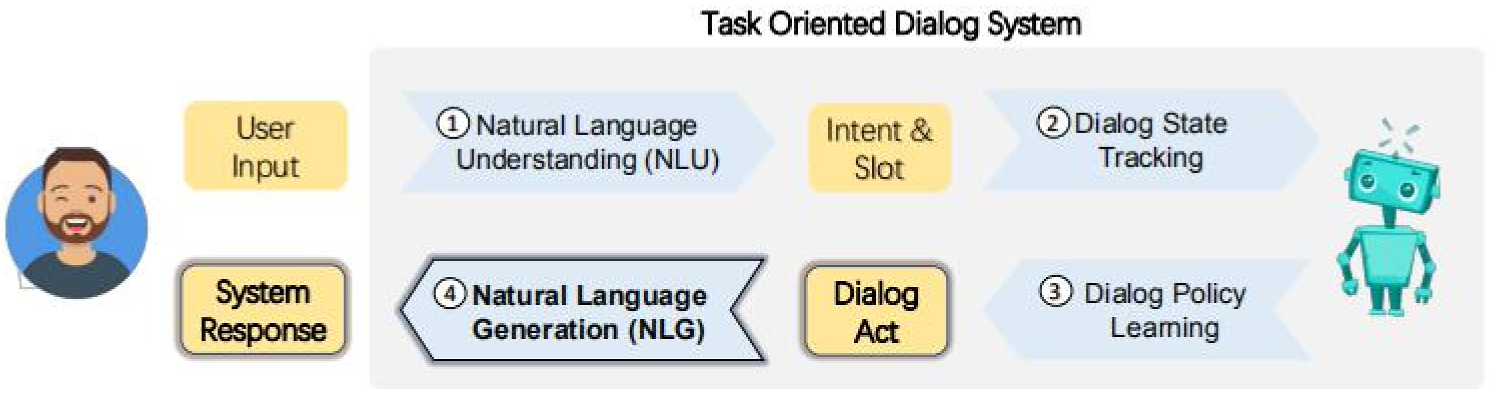
(c) FewRel

(d) HuffPost

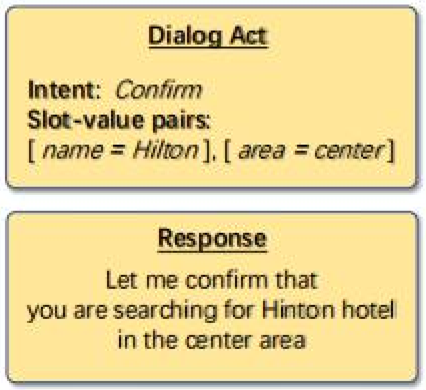
Few-shot Natural Language Generation for Task-Oriented Dialog

Baolin Peng, Chenguang Zhu, Chunyuan Li
Xiujun Li, Jinchao Li, Michael Zeng, Jianfeng Gao
Microsoft Research, Redmond
{bapeng,chezhu,chunyl,xiul,jincli,nzeng,jfgao}@microsoft.com

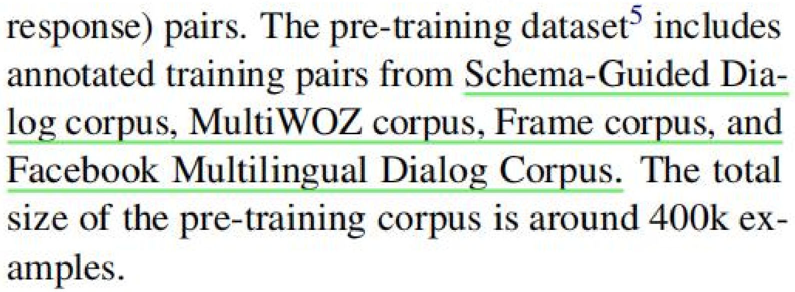
- A new benchmark **FEWSHOTWOZ** is introduced.
- They propose a new model **SC-GPT**.
- On the MultiWOZ dataset, SC-GPT creates a new SOTA.


Task Oriented Dialog System
User Input
System Response
① Natural Language Understanding (NLU)
Intent & Slot
② Dialog State Tracking
④ Natural Language Generation (NLG)
Dialog Act
③ Dialog Policy Learning

(a) The overall framework of a task-oriented dialog system

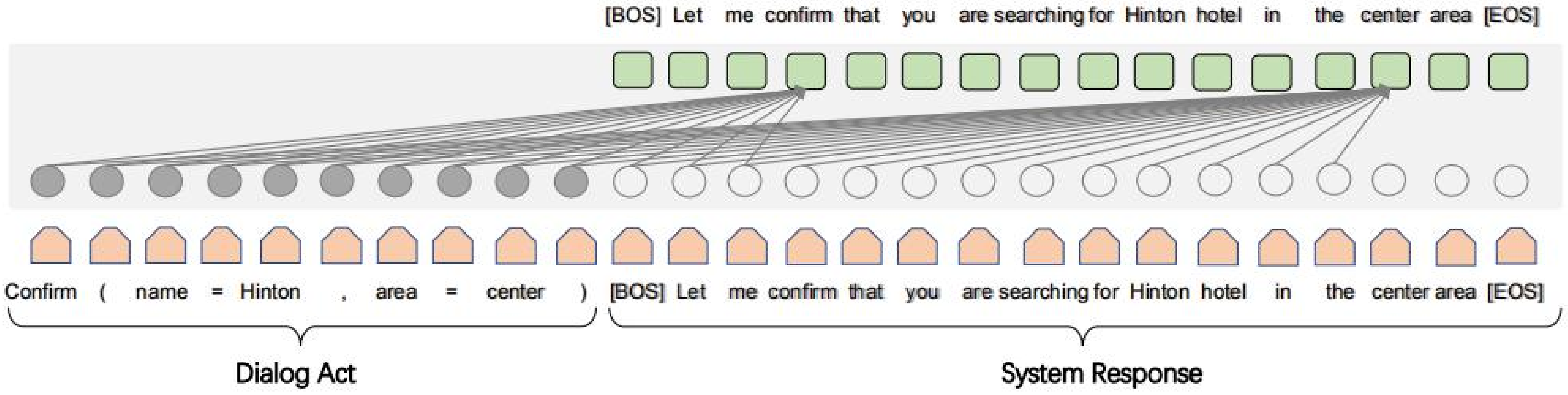


(b) Dialog act & Response



response) pairs. The pre-training dataset[5] includes annotated training pairs from Schema-Guided Dialog corpus, MultiWOZ corpus, Frame corpus, and Facebook Multilingual Dialog Corpus. The total size of the pre-training corpus is around 400k examples.

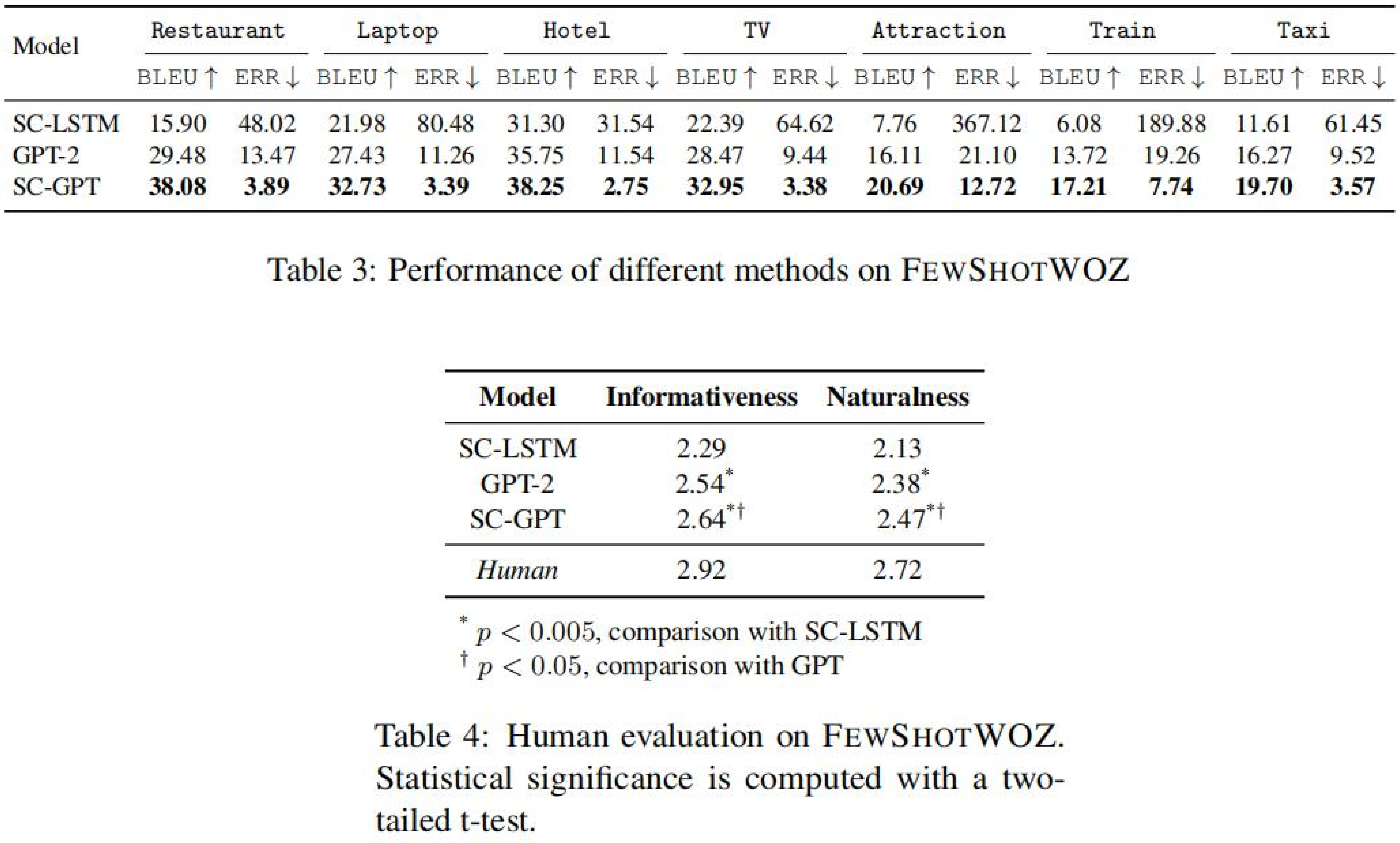
Pre-training + pre-training + fine tuning = better response


[BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Confirm (name = Hinton , area = center) [BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
Dialog Act
System Response

Massive Plain Language Pre-training.

Dialog-Act Controlled Pre-training.

Fine-tuning.



Model	Restaurant		Laptop		Hotel		TV		Attraction		Train		Taxi	
	BLEU ↑	ERR ↓	BLEU ↑	ERR ↓	BLEU ↑	ERR ↓	BLEU ↑	ERR ↓	BLEU ↑	ERR ↓	BLEU ↑	ERR ↓	BLEU ↑	ERR ↓
SC-LSTM	15.90	48.02	21.98	80.48	31.30	31.54	22.39	64.62	7.76	367.12	6.08	189.88	11.61	61.45
GPT-2	29.48	13.47	27.43	11.26	35.75	11.54	28.47	9.44	16.11	21.10	13.72	19.26	16.27	9.52
SC-GPT	**38.08**	**3.89**	**32.73**	**3.39**	**38.25**	**2.75**	**32.95**	**3.38**	**20.69**	**12.72**	**17.21**	**7.74**	**19.70**	**3.57**

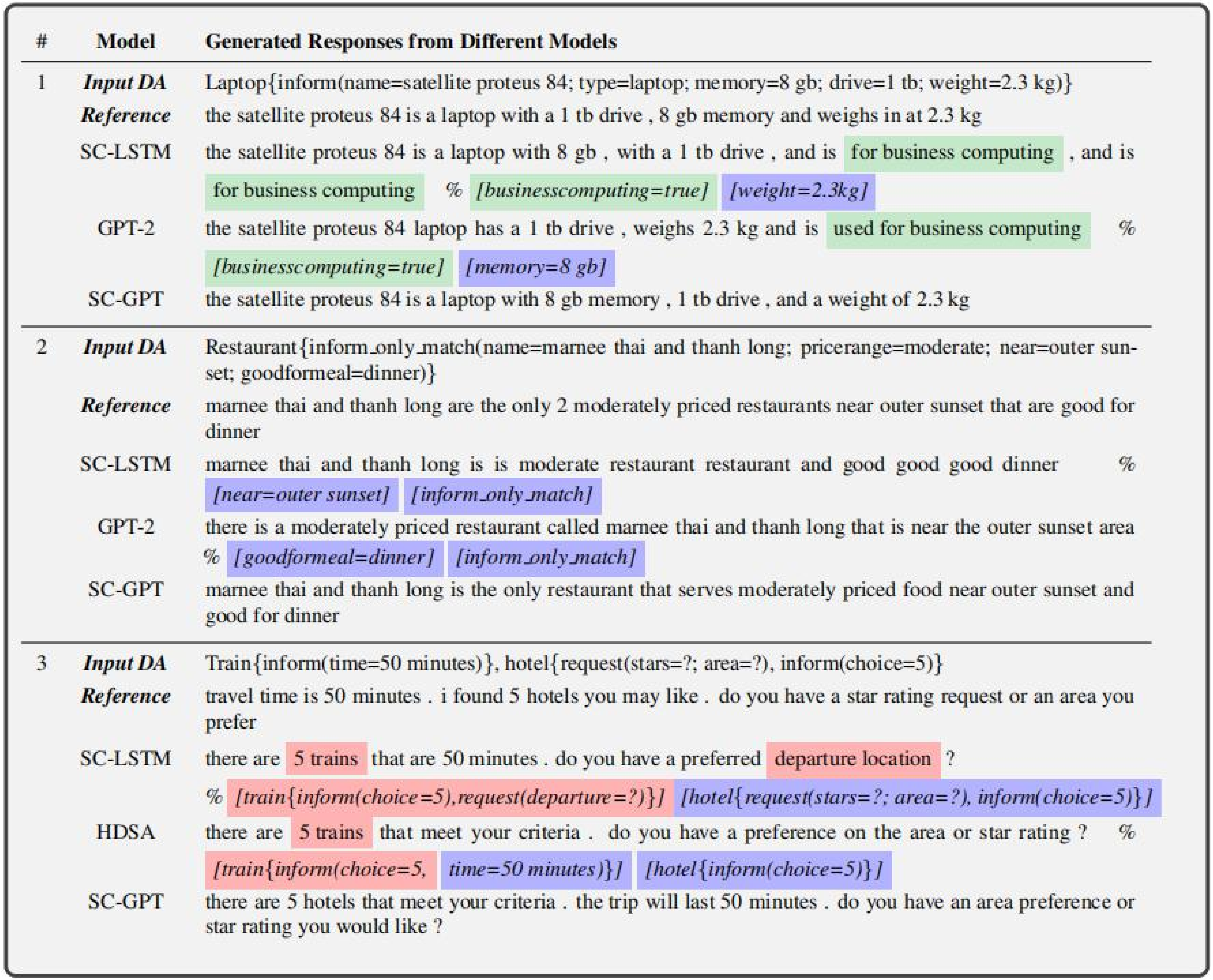
Table 3: Performance of different methods on FEWSHOTWOZ

Model	**Informativeness**	**Naturalness**
SC-LSTM	2.29	2.13
GPT-2	2.54^{*}	2.38^{*}
SC-GPT	$2.64^{*\dagger}$	$2.47^{*\dagger}$
Human	2.92	2.72

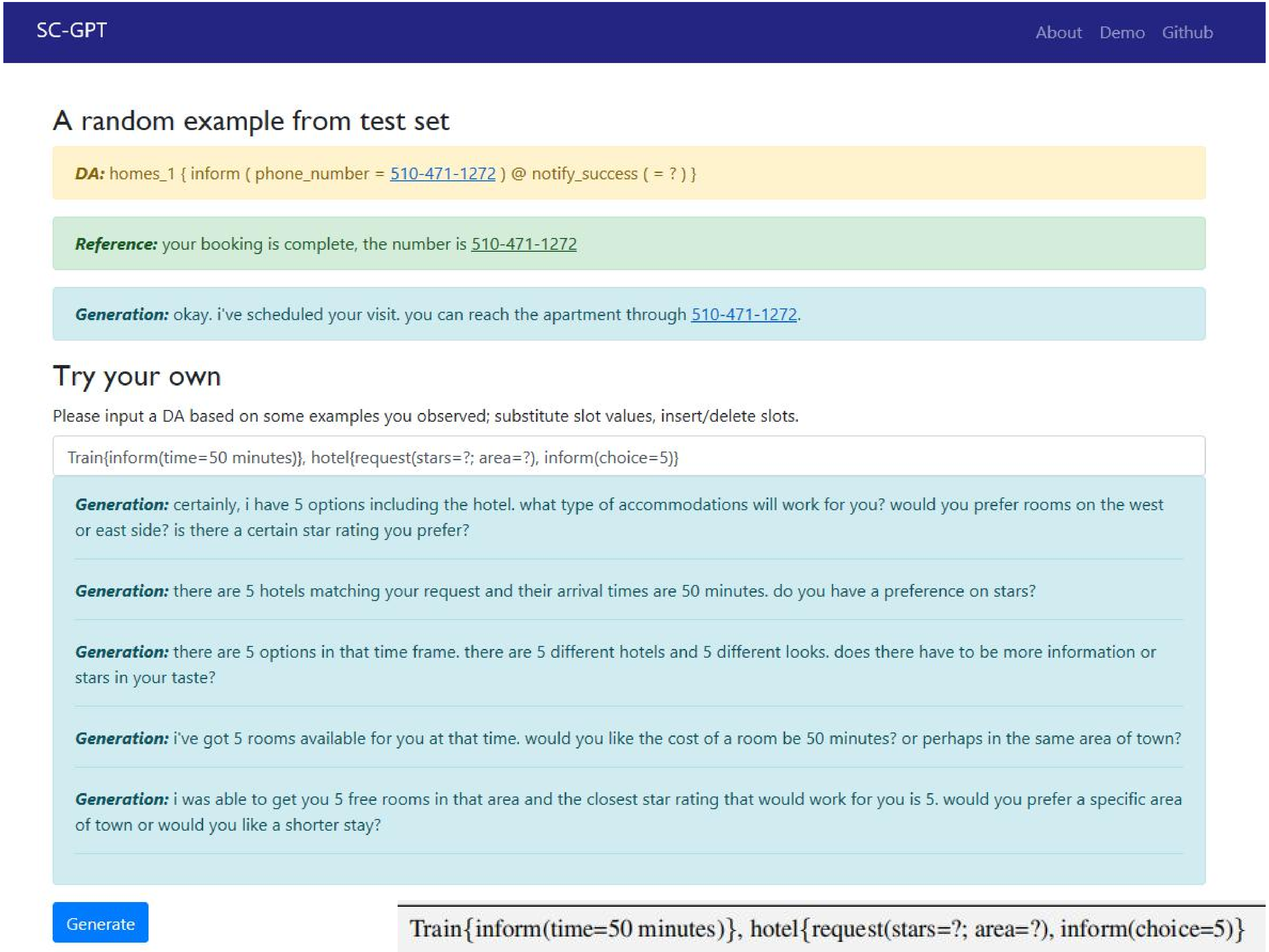
* $p < 0.005$, comparison with SC-LSTM

† $p < 0.05$, comparison with GPT

Table 4: Human evaluation on FEWSHOTWOZ. Statistical significance is computed with a two-tailed t-test.



#	Model	Generated Responses from Different Models
1	***Input DA***	Laptop{inform(name=satellite proteus 84; type=laptop; memory=8 gb; drive=1 tb; weight=2.3 kg)}
	Reference	the satellite proteus 84 is a laptop with a 1 tb drive , 8 gb memory and weighs in at 2.3 kg
	SC-LSTM	the satellite proteus 84 is a laptop with 8 gb , with a 1 tb drive , and is for business computing , and is for business computing % *[businesscomputing=true]* *[weight=2.3kg]*
	GPT-2	the satellite proteus 84 laptop has a 1 tb drive , weighs 2.3 kg and is used for business computing % *[businesscomputing=true]* *[memory=8 gb]*
	SC-GPT	the satellite proteus 84 is a laptop with 8 gb memory , 1 tb drive , and a weight of 2.3 kg
2	***Input DA***	Restaurant{inform_only_match(name=marnee thai and thanh long; pricerange=moderate; near=outer sunset; goodformeal=dinner)}
	Reference	marnee thai and thanh long are the only 2 moderately priced restaurants near outer sunset that are good for dinner
	SC-LSTM	marnee thai and thanh long is is moderate restaurant restaurant and good good good dinner % *[near=outer sunset]* *[inform_only_match]*
	GPT-2	there is a moderately priced restaurant called marnee thai and thanh long that is near the outer sunset area % *[goodformeal=dinner]* *[inform_only_match]*
	SC-GPT	marnee thai and thanh long is the only restaurant that serves moderately priced food near outer sunset and good for dinner
3	***Input DA***	Train{inform(time=50 minutes)}, hotel{request(stars=?; area=?), inform(choice=5)}
	Reference	travel time is 50 minutes . i found 5 hotels you may like . do you have a star rating request or an area you prefer
	SC-LSTM	there are 5 trains that are 50 minutes . do you have a preferred departure location ? % *[train{inform(choice=5),request(departure=?)}]* *[hotel{request(stars=?; area=?), inform(choice=5)}]*
	HDSA	there are 5 trains that meet your criteria . do you have a preference on the area or star rating ? % *[train{inform(choice=5,* *time=50 minutes)}]* *[hotel{inform(choice=5)}]*
	SC-GPT	there are 5 hotels that meet your criteria . the trip will last 50 minutes . do you have an area preference or star rating you would like ?



A random example from test set

DA: homes_1 { inform (phone_number = 510-471-1272) @ notify_success (= ?) }

Reference: your booking is complete, the number is 510-471-1272

Generation: okay. i've scheduled your visit. you can reach the apartment through 510-471-1272.

Try your own

Please input a DA based on some examples you observed; substitute slot values, insert/delete slots.

Train{inform(time=50 minutes)}, hotel{request(stars=?; area=?), inform(choice=5)}

Generation: certainly, i have 5 options including the hotel. what type of accommodations will work for you? would you prefer rooms on the west or east side? is there a certain star rating you prefer?

Generation: there are 5 hotels matching your request and their arrival times are 50 minutes. do you have a preference on stars?

Generation: there are 5 options in that time frame. there are 5 different hotels and 5 different looks. does there have to be more information or stars in your taste?

Generation: i've got 5 rooms available for you at that time. would you like the cost of a room be 50 minutes? or perhaps in the same area of town?

Generation: i was able to get you 5 free rooms in that area and the closest star rating that would work for you is 5. would you prefer a specific area of town or would you like a shorter stay?

Train{inform(time=50 minutes)}, hotel{request(stars=?; area=?), inform(choice=5)}

Thanks