EMNLP 2019

Andrey Bout, Huawei

What is EMNLP 2019 is

- ~2500 participants
- 2876 articles submitted (23.7% acceptance rate)
- 3 days + 2 days of workshops
- e.t.c.

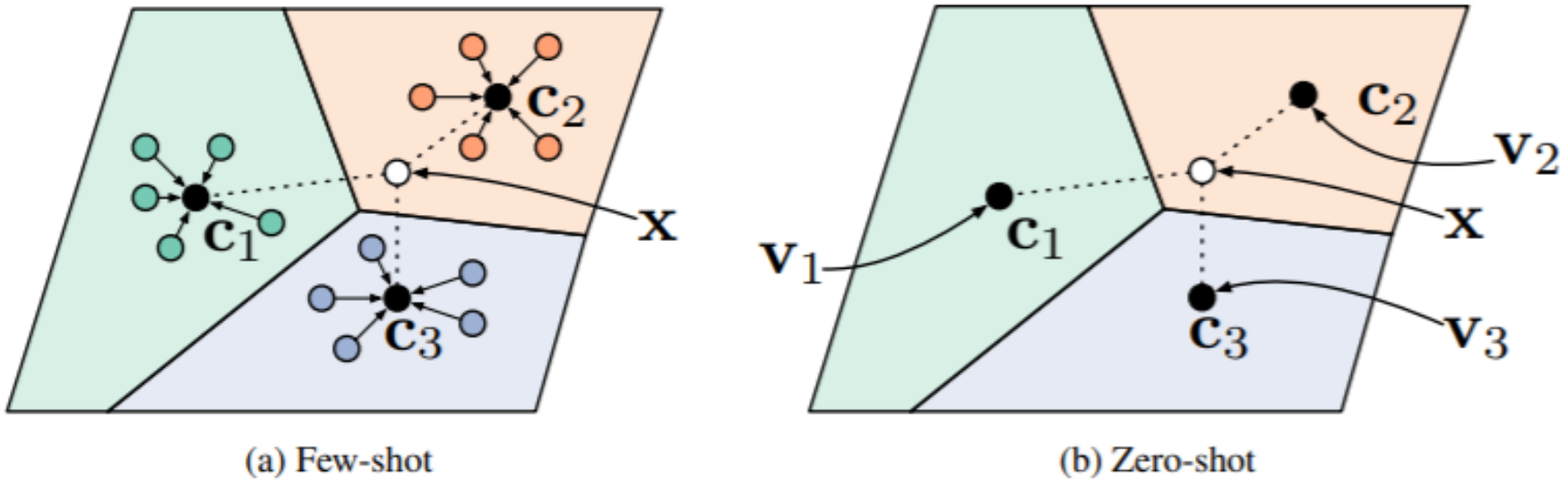
What I am going to outline

- Low-resource approaches
- Dialog systems
- Multilinguality
- Few-shot and zero-shot learning
- Language models

What I have no time to speak about

- Machine translation
- Summarization

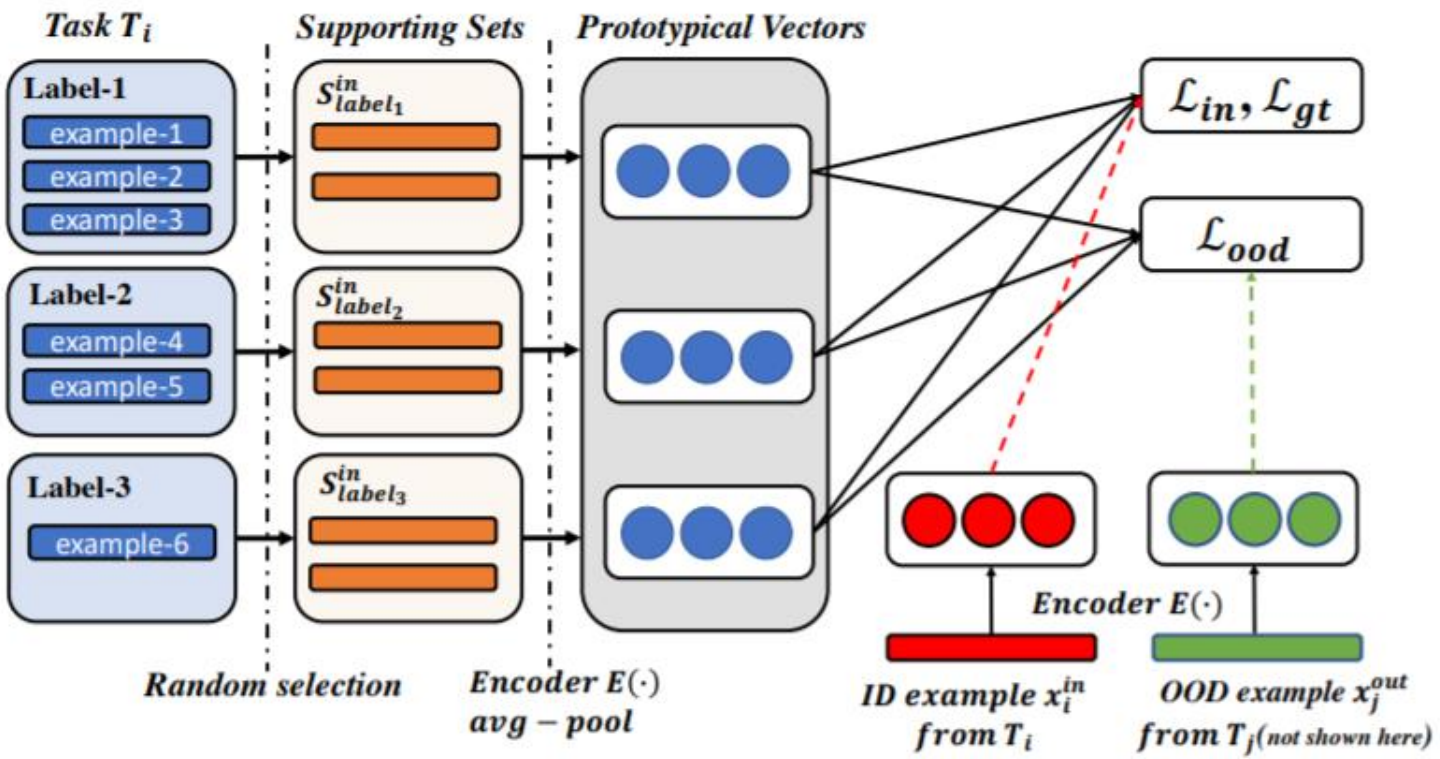
Prototypical network overview



(a) Few-shot

(b) Zero-shot

Out-of-Domain Detection for Low-Resource Text Classification Tasks



$$\mathcal{L}_{in} = -\log \frac{\exp \alpha F(x_i^{in}, S_{l_i}^{in})}{\sum_{l'} \exp \alpha F(x_i^{in}, S_{l'}^{in})}$$

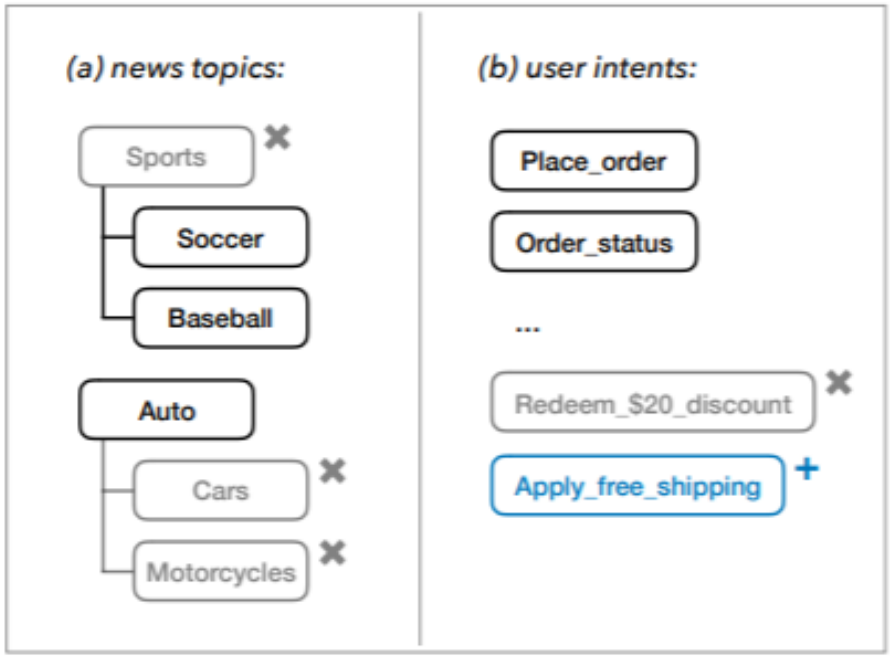
$$\mathcal{L}_{ood} = max[0, \max_{l}(F(x_j^{out}, S_l^{in}) - \mathcal{M}_1)]$$

$$\mathcal{L}_{gt} = max[0, \mathcal{M}_2 - F(x_i^{in}, S_{l_i}^{in}))]$$

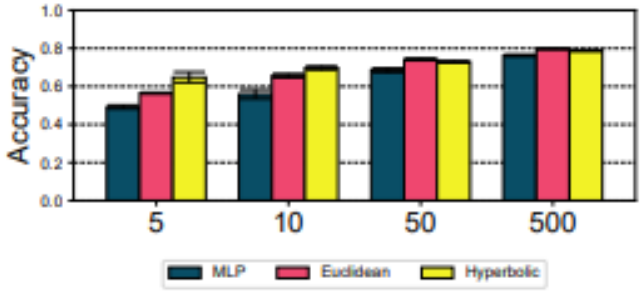
	Conversation			Amazon Review		
(%)	EER	CER	Comb.	EER	CER	Comb.
OSVM	63.6	-	-	47.6	-	-
LSTM AutoEnc.	48.0	78.4	79.5	45.4	29.3	38.6
Vanilla CNN	26.4	76.8	77.6	47.7	34.4	42.8
Proto. Network	26.9	32.5	44.5	46.5	**7.3**	47.6
O-Proto ($\mathcal{L}_{in} + \mathcal{L}_{gt}$)	27.6	33.3	46.2	47.8	7.4	48.9
O-Proto ($\mathcal{L}_{in} + \mathcal{L}_{ood}$)	24.5	30.1	41.2	24.7	9.7	30.1
O-Proto (all)	**24.1**	**29.6**	**40.8**	**24.0**	9.1	**29.1**
Proto. with bilstm	25.0	32.5	42.6	45.1	**6.8**	46.0
O-Proto with bilstm	**22.0**	**30.5**	**39.8**	**21.9**	9.0	**27.1**

Table 2: O-Proto is compared with other baselines for Conversation and Amazon data

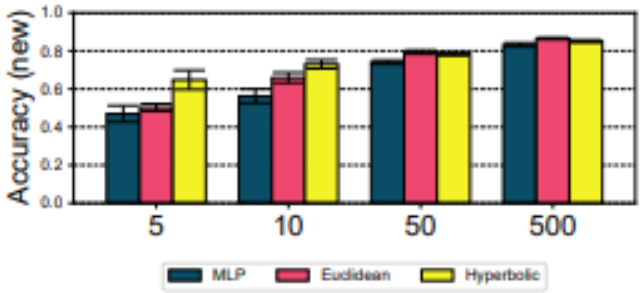
Metric Learning for Dynamic Text Classification



Dataset	Model	$n_{fine} = 5$	$n_{fine} = 10$	$n_{fine} = 20$	$n_{fine} = 100$
SENT	MLP	37.3 ± 2.9	43.8 ± 3.5	45.7 ± 3.8	57.4 ± 3.5
	EUC	39.6 ± 6.4	45.5 ± 1.8	47.7 ± 4.7	$\mathbf{62.7 \pm 2.1}$
	HYP	$\mathbf{42.2 \pm 3.5}$	$\mathbf{47.1 \pm 4.8}$	$\mathbf{53.0 \pm 2.3}$	$\mathbf{62.7 \pm 2.2}$
NEWS	MLP	49.2 ± 1.0	55.9 ± 2.5	68.5 ± 1.1	76.3 ± 0.5
	EUC	56.5 ± 0.4	65.6 ± 1.0	$\mathbf{74.2 \pm 0.6}$	$\mathbf{79.8 \pm 0.2}$
	HYP	$\mathbf{64.8 \pm 2.8}$	$\mathbf{69.7 \pm 1.0}$	72.9 ± 0.5	78.8 ± 0.4
WOS	MLP	36.6 ± 1.1	46.8 ± 1.2	62.8 ± 0.6	68.9 ± 0.5
	EUC	49.4 ± 1.0	59.2 ± 0.4	$\mathbf{70.4 \pm 0.4}$	73.3 ± 0.2
	HYP	$\mathbf{54.5 \pm 1.4}$	$\mathbf{60.7 \pm 0.9}$	70.2 ± 0.7	$\mathbf{73.5 \pm 0.5}$



(a) Accuracy with respect to the full label set



(b) Accuracy with respect to new classes only

Figure 3: Accuracy on the NEWS Dataset against number of fine tune examples: (a) all classes and (b) newly introduced classes only. The mean is taken over 5 random label splits, and error bars are given at ± 1 standard deviation.

Domain Adaptation with BERT-based Domain Classification and Data Selection

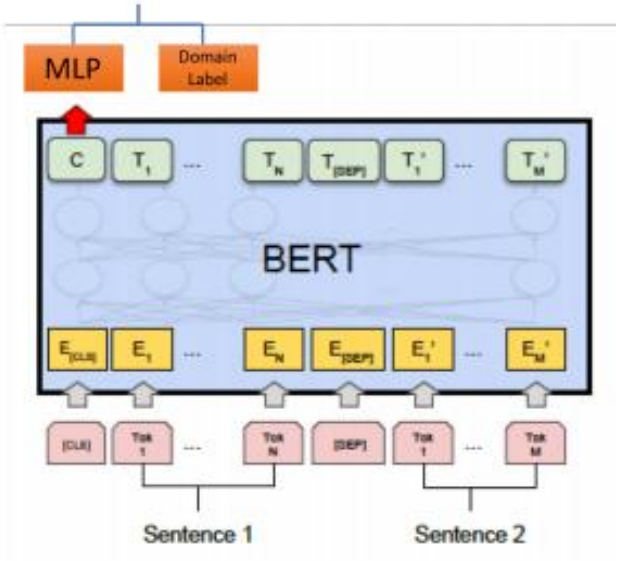


Figure 1: Setup for training a BERT domain classifier. Picture adapted from (Devlin et al., 2018)

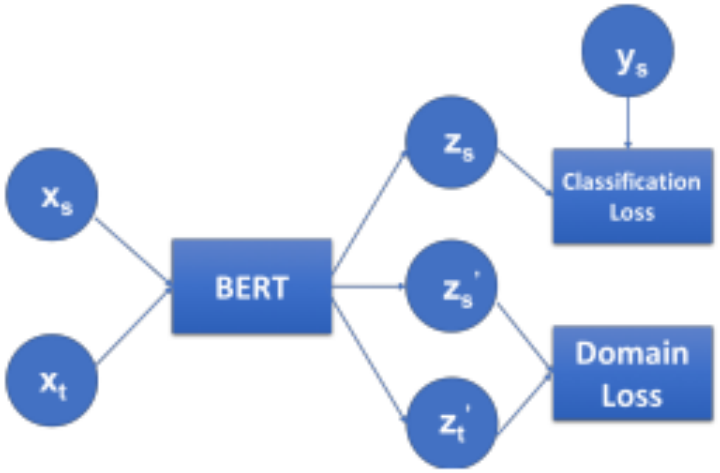


Figure 2: Setup for BERT domain adaptation with MMD-based domain regularization.

Source	Target	IFT	NZS	MMD	DDS	% Data
MNLI	QNLI	85.3	49.8	58.0	**58.5**	0.1 %
MNLI	Quora	89.3	73.7	71.5	**73.9**	26.1 %
MNLI	SNLI	92.9	87.4	87.6	**88.3**	26.1 %
QNLI	MNLI	88.3	63.7	66.0	**67.2**	0.4 %
QNLI	Quora	89.3	61.8	66.1	**67.6**	1.5 %
QNLI	SNLI	92.9	56.1	65.3	**66.6**	0.4 %
Quora	MNLI	88.3	71.0	70.6	**83.6**	3.5 %
Quora	QNLI	85.3	50.8	58.8	**59.1**	1.8 %
Quora	SNLI	92.9	69.1	**72.4**	71.6	1.8 %
SNLI	MNLI	88.3	77.0	**82.2**	80.2	5.0 %
SNLI	QNLI	85.3	49.0	54.9	**56.7**	0.1 %
SNLI	Quora	89.3	67.0	70.8	**70.9**	1.3 %

Table 2: Transfer performance (accuracy) of different domain adaptation methods. "IFT": in-domain fine-tuning. "NZS": naive zero-shot. "MMD": MMD-based domain regularization. "DDS": discriminative data selection. "% Data": percentage of source domain data selected in DDS method.

Evaluating Lottery Tickets Under Distributional Shifts

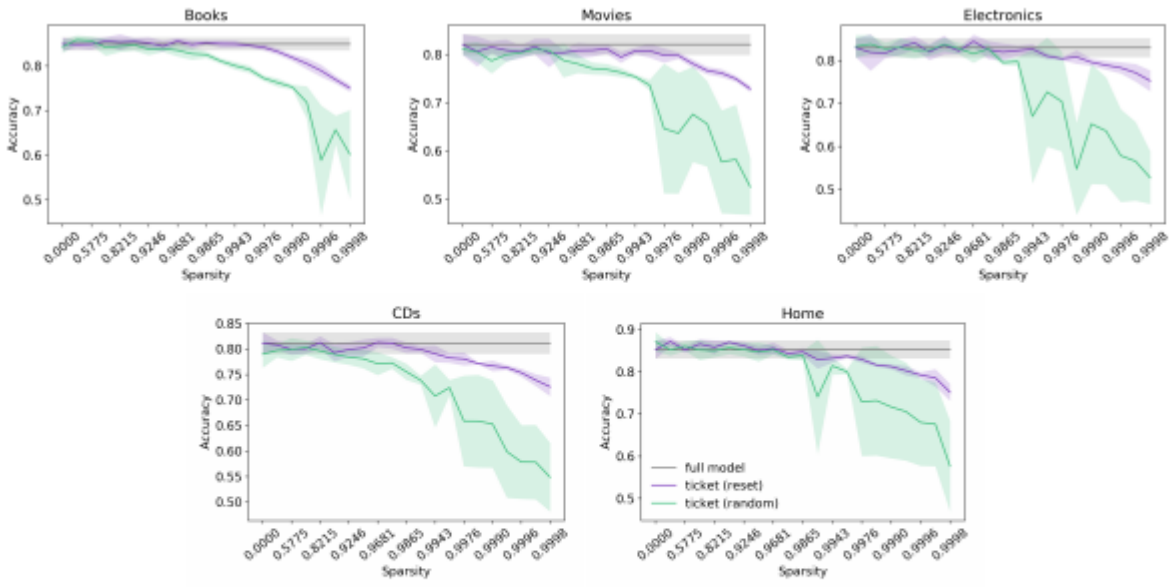


Figure 3: Results obtaining lottery tickets on the Books, Movies, Electronics, CDs, and Home categories of the Amazon Reviews dataset (McAuley and Leskovec, 2013). Experiments are repeated five times, where the solid lines represent the mean and shaded regions represent the standard deviation. Note that the x-axis ticks are *not* uniformly spaced.

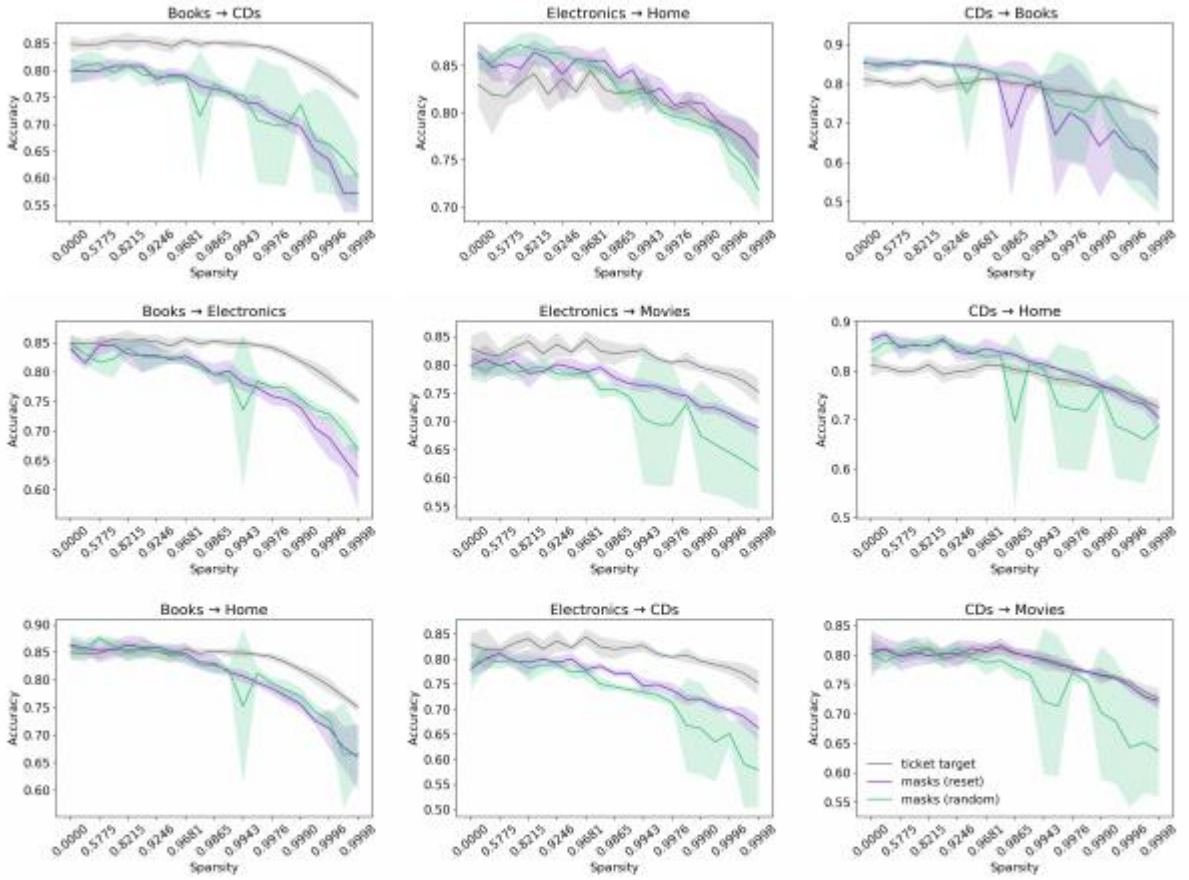
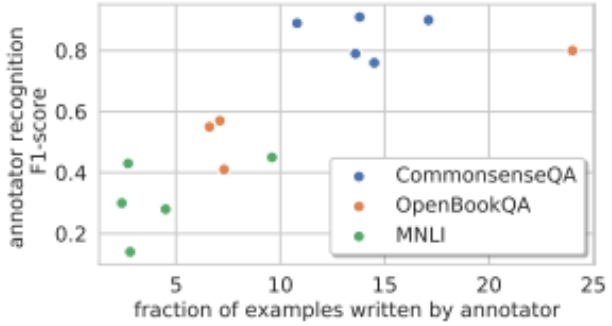


Figure 4: Results transferring lottery tickets on nine transfer tasks constructed from the five categories of the Amazon Reviews dataset (McAuley and Leskovec, 2013). Experiments are repeated five times, where the solid lines represent the mean and shaded regions represent the standard deviation. Note that the x-axis ticks are *not* uniformly spaced.

Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets

	Without ID	With ID	p-value
OpenBookQA	52.2	56.4	$1.83e^{-2}$
CommonsenseQA	53.6	55.3	$11.98e^{-2}$
MNLI	82.9	84.5	$5.13e^{-7}$



CommonsenseQA-single		CommonsenseQA-multi	
4.2 ± 0.7	17.1%	-9.5 ± 8.3	0.9%
7.7 ± 1.9	14.5%	6.5 ± 7.0	0.6%
-2.8 ± 1.3	13.8%	-6.1 ± 8.5	0.5%
-3.8 ± 0.9	13.6%	1.6 ± 10.8	0.4%
1.6 ± 2.7	10.8%	1.8 ± 10.5	0.4%
OpenBookQA-single		**OpenBookQA-multi**	
-0.9 ± 2.7	24%	-14.7 ± 6.2	2.4%
-13.5 ± 1.7	7.8%	-19.4 ± 8.5	1.7%
-5.8 ± 0.7	7.3%	-12.4 ± 5.5	1.2%
8.2 ± 5.2	7.1%	-13.7 ± 8.5	1%
3.1 ± 1.1	6.6%	-23.3 ± 7.8	0.8%
MNLI-single		**MNLI-multi**	
-2.5 ± 0.5	9.6%	2.5 ± 0.8	1.8%
-3.0 ± 0.6	4.5%	-1.1 ± 0.9	1.5%
2.9 ± 0.2	2.8%	-4.6 ± 0.8	1.5%
0.8 ± 0.7	2.7%	-1.5 ± 0.2	1.5%
4.6 ± 0.2	2.4%	0.5 ± 0.2	1.5%

Table 3: Performance difference (p.d.) between single- and multi- annotator splits and random splits of identical size. Each cell shows the p.d. mean and standard deviation, as well as the development set relative size.

To Annotate or Not? Predicting Performance Drop under Domain Shift

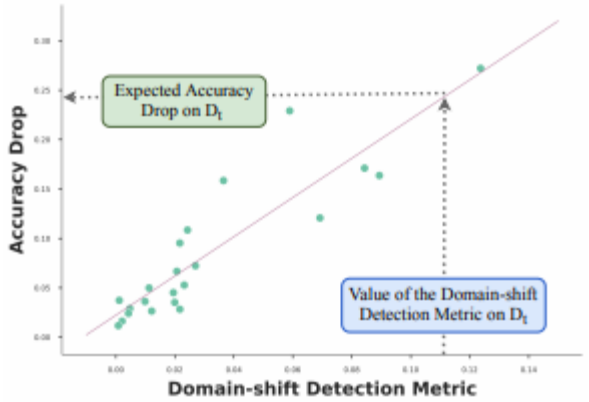


Figure 1: In this paper we introduce several domain-shift detection metrics (x-axis) and employ them to estimate the performance drop on a new target domain D_t by regressing on those metrics and their associated real performance drop (green dots).

	Sentiment		POS tagging	
	MAE	Max	MAE	Max
Mean	5.2 ± 2.02	12.77	1.06 ± 0.36	1.67
RCA	2.88 ± 1.31	7.17	1.08 ± 0.31	1.58
RCA*	2.92 ± 1.39	7.42	1.05 ± 0.27	1.42
CONF	2.85 ± 1.69	8.86	$\mathbf{0.89} \pm 0.39$	1.75
CONF_CALIB	2.67 ± 1.49	8.13	1.12 ± 0.45	1.83
PAD	2.51 ± 1.54	8.16	1.24 ± 0.37	1.7
PAD*	$\mathbf{2.15} \pm \mathbf{0.88}$	**4.64**	$\mathbf{0.89} \pm \mathbf{0.1}$	**0.99**
Ensemble	2.22 ± 0.9	5.02	1.03 ± 0.42	1.78

Measure	Robust to co-variate shift	Black box	Good with small Target Data
PAD	X	✓	X
PAD*	(X)	X	X
RCA	✓	✓	X
RCA*	✓	✓	X
CONF	✓	X	✓
CONF_CALIB	✓	X	✓

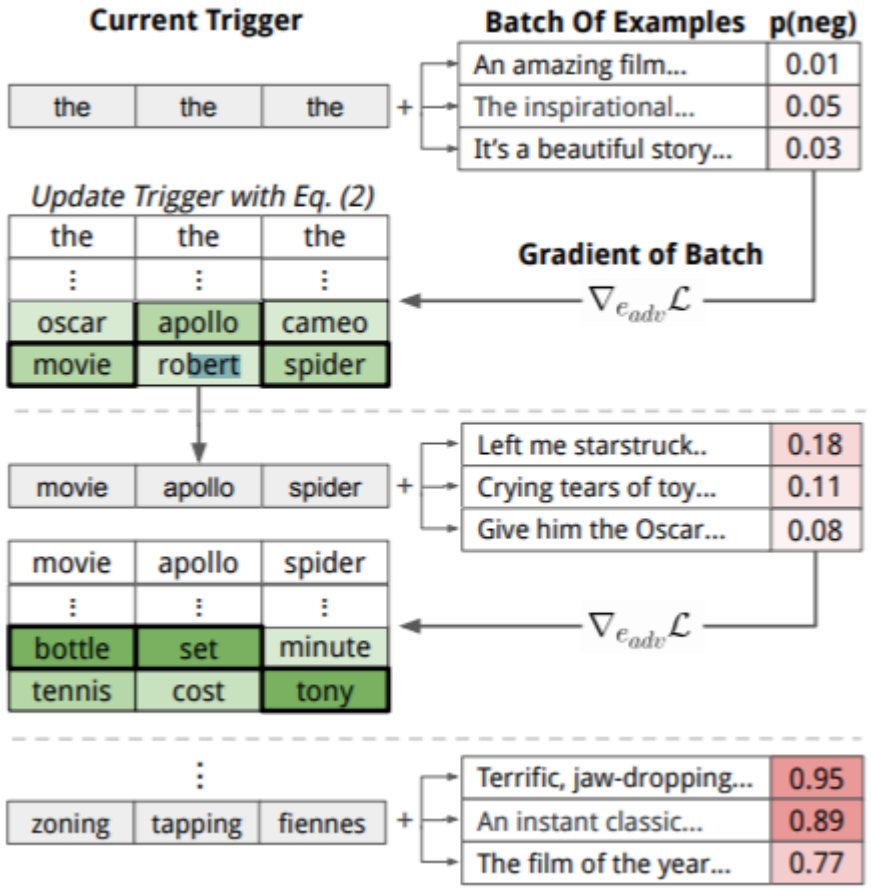
Table 3: Summary of domain similarity metrics discussed in this work and their different characteristics.

	MAE	max
Mean	4.35 ± 2.31	10.79
RCA	3.63 ± 1.95	9.56
RCA*	3.67 ± 1.93	8.44
CONF	2.88 ± 1.16	5.5
CONF_CALIB	$\mathbf{2.81} \pm \mathbf{1.09}$	**5.48**
PAD	4.35 ± 2.31	10.79
PAD*	4.6 ± 2.14	10.77
Ensemble	3.02 ± 0.98	5.88

Table 2: Mean absolute error and Max error of performance drop prediction for sentiment analysis under adversarial shift evaluation (The lower the better).

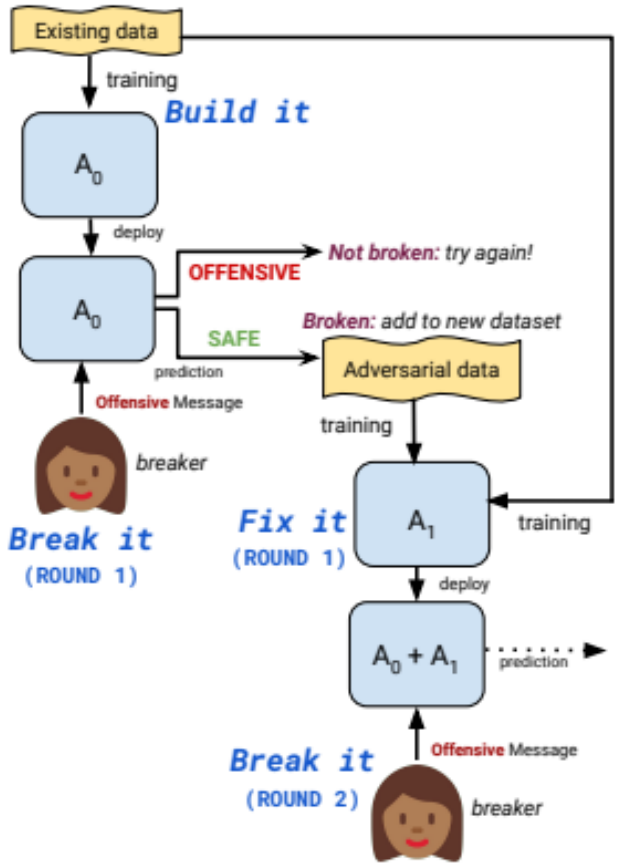
https://github.com/hadyelsahar/domain-shift-prediction

Universal Adversarial Triggers for Attacking and Analyzing NLP



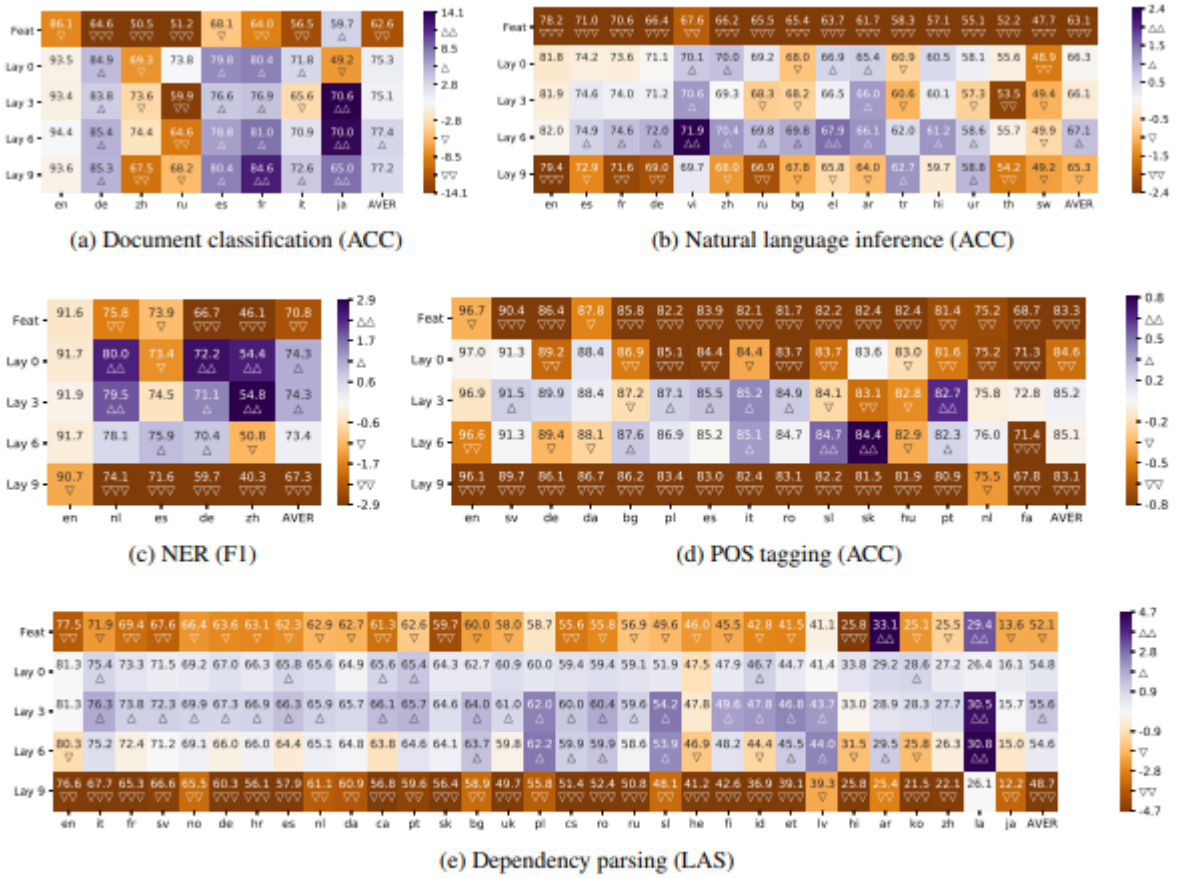
Ground Truth	Trigger	ESIM	DA	DA-ELMo
		89.49	89.46	90.88
	nobody	0.03	0.15	0.50
	never	0.50	1.07	0.15
Entailment	sad	1.51	0.50	0.71
	scared	1.13	0.74	1.01
	championship	0.83	0.06	0.77
	Avg. Δ	-88.69	-88.96	-90.25
		84.62	79.71	83.04
	nobody	0.53	8.45	13.61
Neutral	sleeps	4.57	14.82	22.34
	nothing	1.71	23.61	14.63
	none	5.96	17.52	15.41
	sleeping	6.06	15.84	28.86
	Avg. Δ	-80.85	-63.66	-64.07
		86.31	84.80	85.17
	joyously	73.31	70.93	60.67
Contradiction	anticipating	79.89	66.91	62.96
	talented	79.83	65.71	64.01
	impress	80.44	63.79	70.56
	inspiring	78.00	65.83	70.56
	Avg. Δ	-8.02	-18.17	-19.42

Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack



		WTC Baseline	*Standard* models			*Adversarial* models		
Task Type	Task Round	A_0	S_1	S_2	S_3	A_1	A_2	A_3
WTC	-	**83.3**	80.6	81.1	82.1	81.3	78.9	78.0
Standard Task	All (1-3)	68.1	83.3	85.8	**88.0**	83.0	85.3	83.7
Adversarial Task	1	0.0	51.7	69.3	68.6	71.8	**79.0**	78.2
	2	0.0	10.8	26.4	31.8	0.0	**64.4**	62.1
	3	0.0	12.3	17.1	13.7	32.1	0.0	**59.9**
	All (1-3)	0.0	27.4	41.7	41.8	40.6	55.5	**67.6**

Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT



(a) Document classification (ACC)

(b) Natural language inference (ACC)

(c) NER (F1)

(d) POS tagging (ACC)

(e) Dependency parsing (LAS)

Figure 1: Performance of different fine-tuning approaches compared with fine-tuning all mBERT parameters. Color denotes absolute difference and number in each entry is the evaluation in the corresponding setting. Languages are sorted by mBERT zero-shot transfer performance. Three downward triangles indicate performance drop more than the legends lower limit.

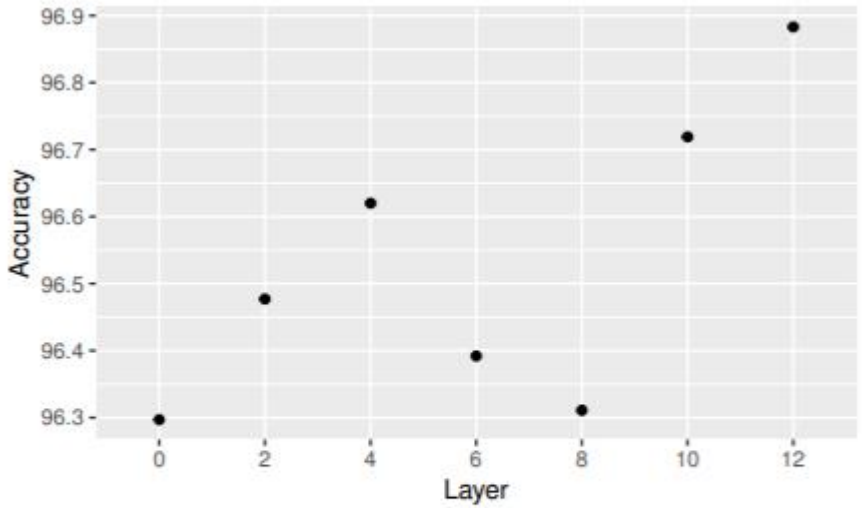


Figure 2: Language identification accuracy for different layer of mBERT. layer 0 is the embedding layer and the layer $i > 0$ is output of the i^{th} transformer block.

BERT is Not an Interlingua and the Bias of Tokenization

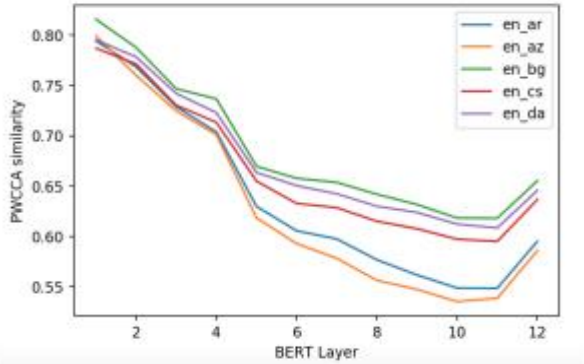


Figure 3: The similarity between representations of different languages decreases deeper into a pretrained uncased multilingual BERT model. Here we show the similarity between English and 5 other languages as a function of model depth

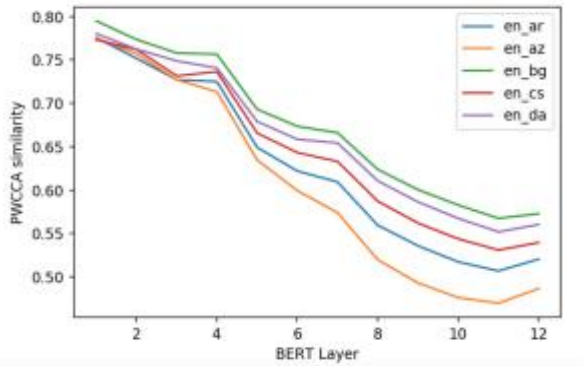


Figure 4: The similarity between representations of different languages decreases deeper into an uncased multilingual BERT model finetuned on XNLI.

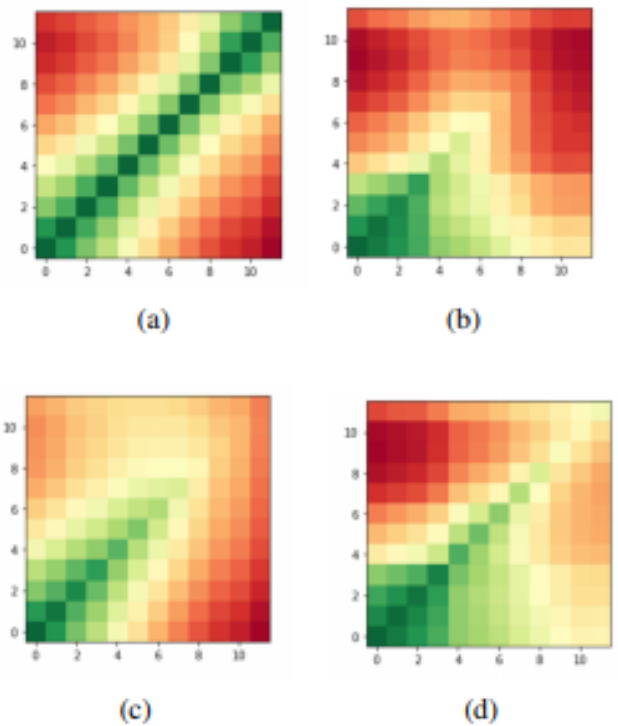


Figure 5: CCA experiments showing the finetuning behavior of BERT

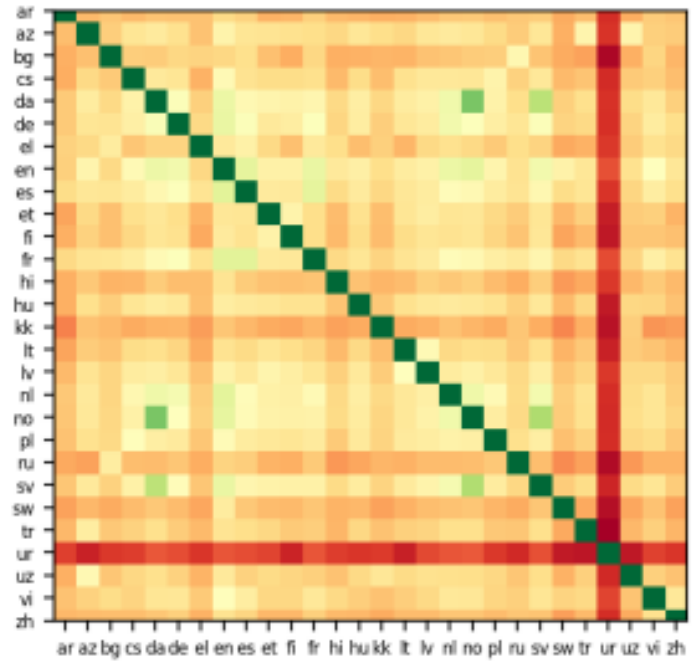


Figure 6: PWCCA generated similarity matrix between languages.

Adaptively Sparse Transformers

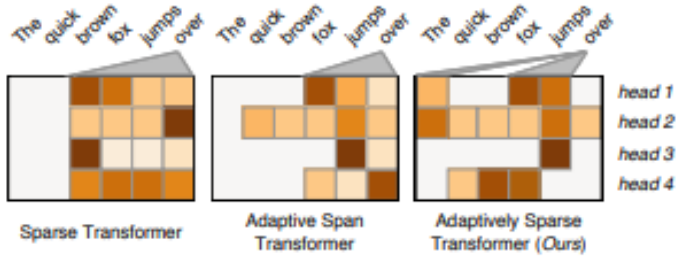
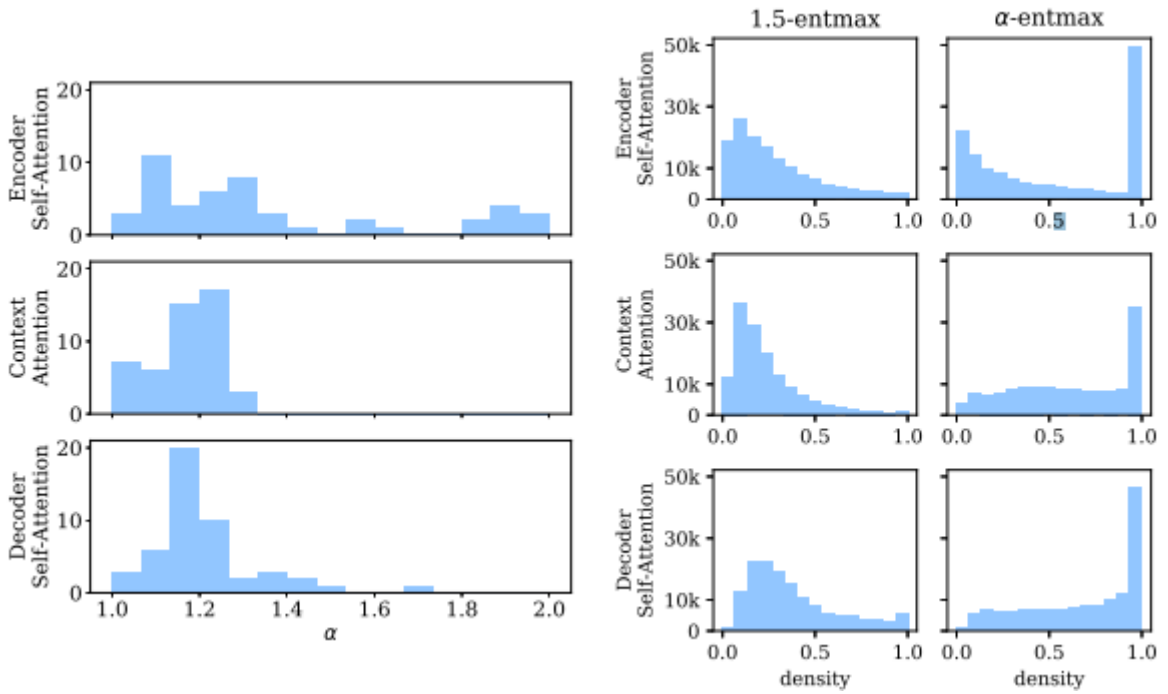


Figure 1: Attention distributions of different self-attention heads for the time step of the token "over", shown to compare our model to other related work. While the sparse Transformer (Child et al., 2019) and the adaptive span Transformer (Sukhbaatar et al., 2019) only attend to words within a contiguous span of the past tokens, our model is not only able to obtain different and not necessarily contiguous sparsity patterns for each attention head, but is also able to tune its support over which tokens to attend adaptively.

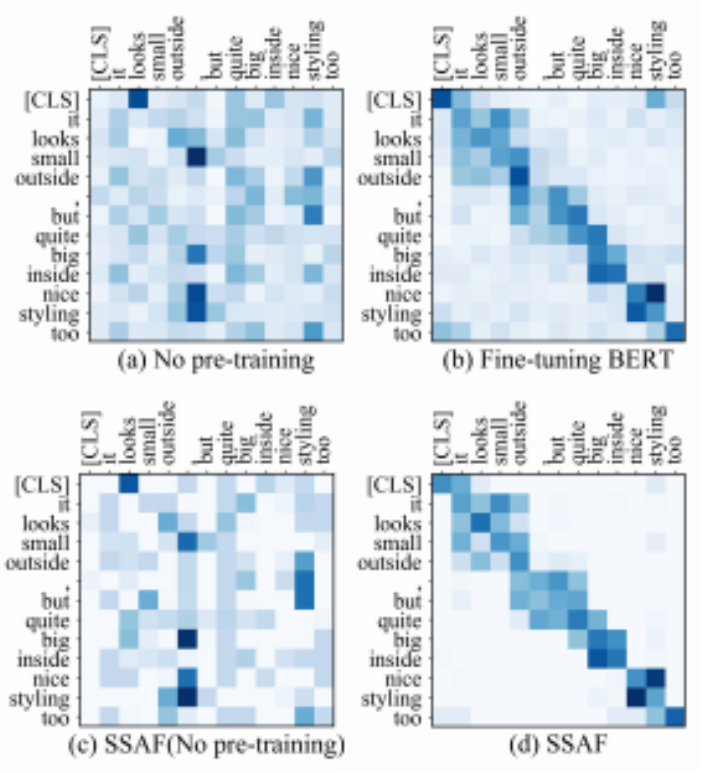
$$\alpha\text{-entmax}(\boldsymbol{z}) = [(\alpha - 1)\boldsymbol{z} - \tau\mathbf{1}]_+^{1/\alpha-1}, \quad (6)$$

activation	DE→EN	JA→EN	RO→EN	EN→DE
softmax	29.79	21.57	32.70	26.02
1.5-entmax	29.83	**22.13**	**33.10**	25.89
α-entmax	**29.90**	21.74	32.89	**26.93**

Table 1: Machine translation tokenized BLEU test results on IWSLT 2017 DE→EN, KFTT JA→EN, WMT 2016 RO→EN and WMT 2014 EN→DE, respectively.



Fine-tune BERT with Sparse Self-Attention Mechanism



Models	SST-1	SST-2	SenTube-A	SenTube-T	SemEval
Ave	42.3	81.1	61.5	64.3	63.6
LSTM	46.5	82.9	57.4	63.6	67.6
BiLSTM	47.1	83.7	59.3	66.2	65.1
CNN	41.9	81.8	57.3	62.1	63.5
SSAN	48.6	85.3	62.5	68.4	72.2
Np	50.1	85.2	66.8	69.6	70.0
SSAF(Np)	50.7	86.4	68.1	70.3	71.2
$\text{BERT}_{\text{BASE}}$	55.2	93.5	70.3	73.3	76.2
SSAF	**56.2**	**94.7**	**72.4**	**75.0**	**77.3**

Table 2: Experimental results of classification accuracy for different methods with five datasets on sentiment analysis task.

Models	SQuAD		SciTail
	EM	F1	Accuracy
Np	65.3	74.1	78.6
SSAF(Np)	66.2	74.6	78.9
$\text{BERT}_{\text{BASE}}$	80.8	88.5	92.0
SSAF	**81.6**	**88.8**	**93.5**

Table 3: Experimental results on question answering and natural language inference tasks.

Small and Practical BERT Models for Sequence Labeling

Input	32 words	128 words
Relative Speedup on GPU		
Meta-LSTM	0.8x	0.2x
MiniBERT	**4.3x**	**2.6x**
Relative Speedup on CPU		
Meta-LSTM	6.8x	2.3x
MiniBERT	**27.7x**	**14.0x**

Model	Part-of-Speech F1	Morphology F1
*m*Meta-LSTM	91.1	82.9
*m*MiniBERT	93.7	88.6
*m*BERT	**94.5**	**91.0**

	POS Tagging						Morphology					
Languages	kk	hy	lt	be	mr	ta	kk	hy	lt	be	mr	ta
Train Size	31	50	153	260	373	400	31	50	153	260	373	400
Meta-LSTM	61.7	75.4	81.4	91.1	72.1	72.7	48.5	54.5	69.7	74.0	59.1	71.0
BERT	75.9	84.4	88.9	94.8	**77.5**	**75.7**	47.8	44.8	**75.2**	82.8	64.0	72.9
*m*BERT	**81.4**	**86.6**	**90.0**	**95.0**	75.9	74.3	**64.6**	**51.1**	73.6	**87.5**	**64.2**	**73.8**
*m*Meta-LSTM	52.9	63.8	65.6	87.6	65.5	61.5	25.6	36.6	42.5	59.2	33.6	46.9
*m*MiniBERT	**76.6**	**86.0**	**86.9**	**95.0**	**75.4**	**74.6**	**59.7**	**47.6**	**64.8**	**81.6**	**59.4**	**71.7**

Revealing the Dark Secrets of BERT

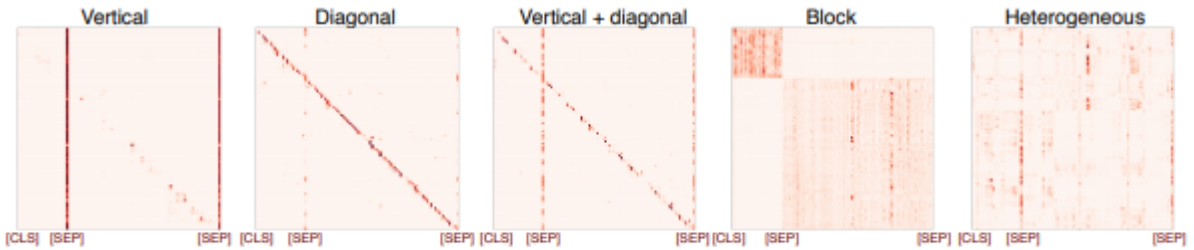


Figure 1: Typical self-attention classes used for training a neural network. Both axes on every image represent BERT tokens of an input example, and colors denote absolute attention weights (darker colors stand for greater weights). The first three types are most likely associated with language model pre-training, while the last two potentially encode semantic and syntactic information.

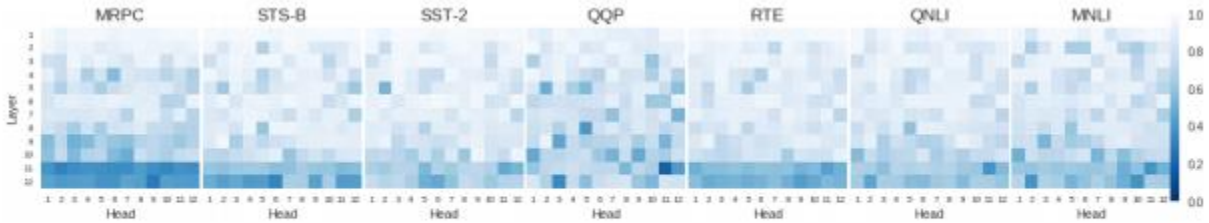


Figure 5: Per-head cosine similarity between pre-trained BERT's and fine-tuned BERT's self-attention maps for each of the selected GLUE tasks, averaged over validation dataset examples. Darker colors correspond to greater differences.

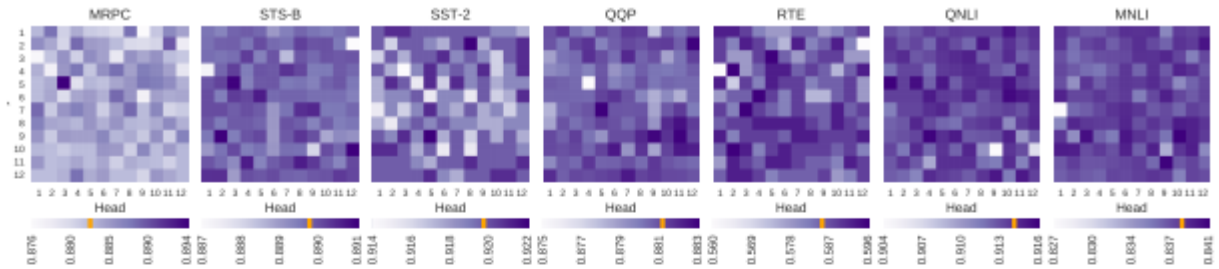


Figure 8: Performance of the model while disabling one head at a time. The orange line indicates the baseline performance with no disabled heads. Darker colors correspond to greater performance scores.

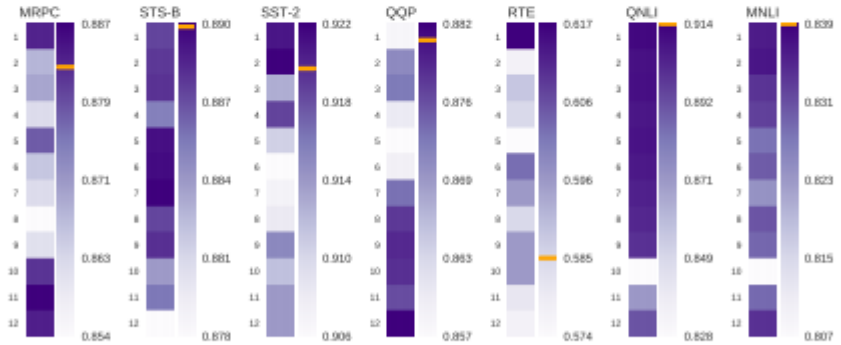
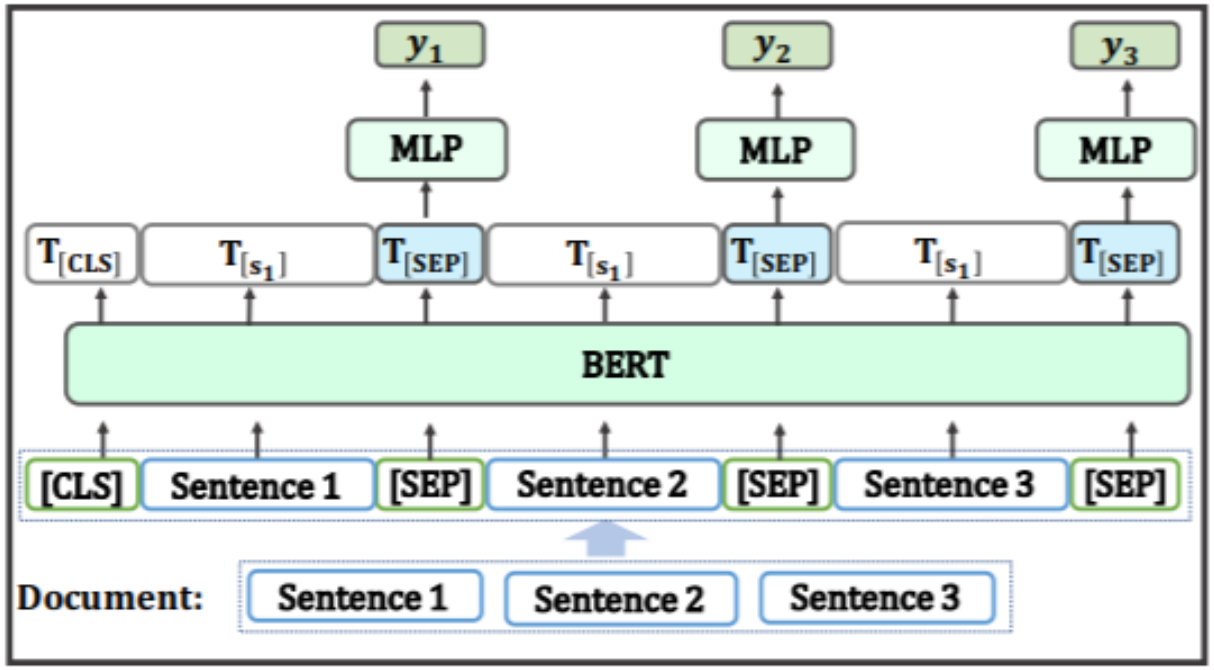


Figure 9: Performance of the model while disabling one layer (that is, all 12 heads in this layer) at a time. The orange line indicates the baseline performance with no disabled layers. Darker colors correspond to greater performance scores.

Pretrained Language Models for Sequential Sentence Classification



Model	PUBMED	CSABST.	NICTA
Jin and Szolovits (2018)	92.6	81.3	84.7
BERT +Transformer	89.6	78.8	78.4
BERT +Transformer+CRF	92.1	78.5	79.1
Our model	**92.9**	**83.1**	**84.8**

Table 3: Abstract sentence classification (micro F1).

Model	ROUGE-L
SAF + F Ens (Collins et al., 2017)	0.313
BERT +Transformer	0.287
Our model	0.306
Our model + ABSTRACTROUGE	**0.314**

Table 4: Results on CSPUBSUM

Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

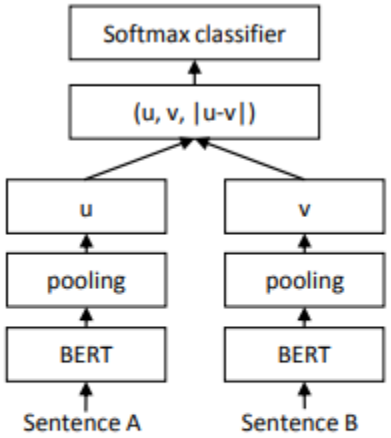


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

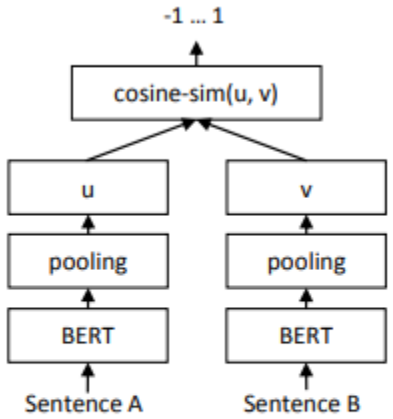


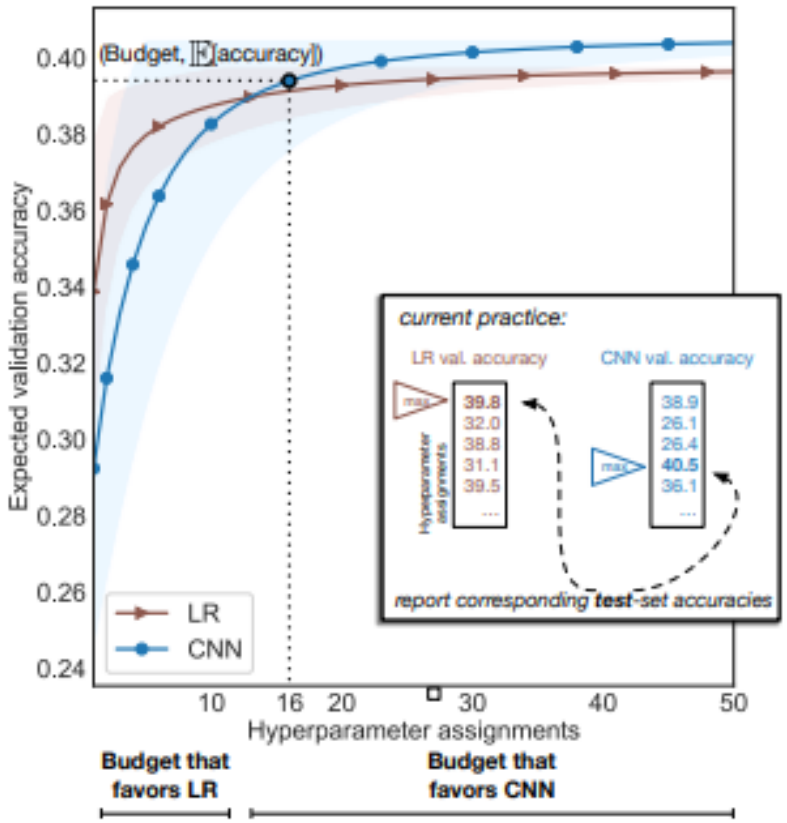
Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

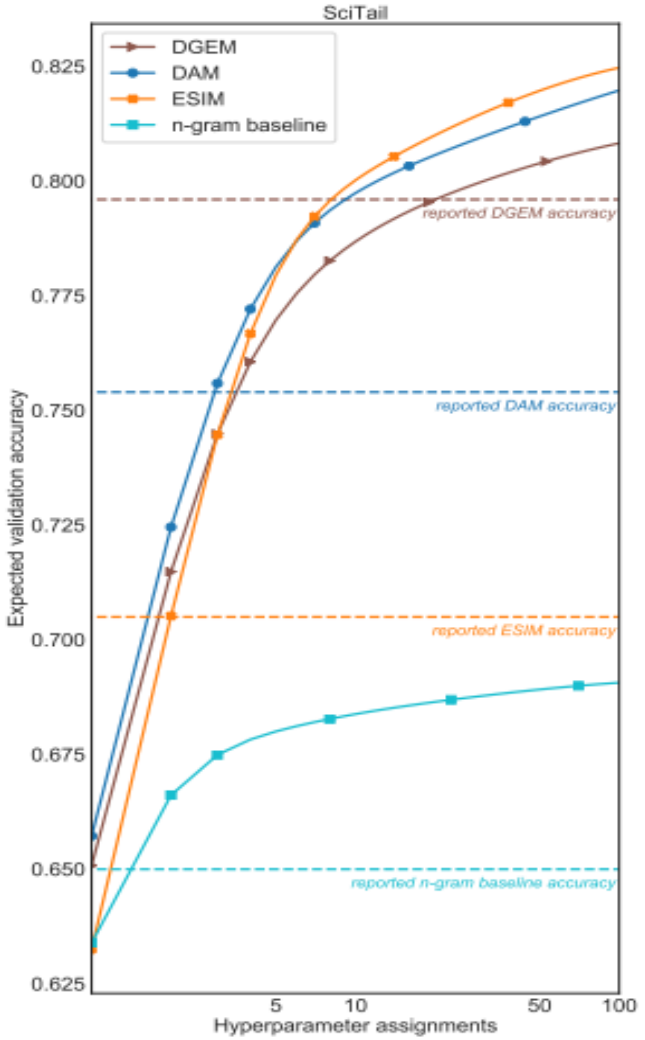
Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Avg. GloVe embeddings	77.25	78.30	91.17	87.85	80.18	83.0	72.87	81.52
Avg. fast-text embeddings	77.96	79.23	91.68	87.81	82.15	83.6	74.49	82.42
Avg. BERT embeddings	78.66	86.25	94.37	88.66	84.40	92.8	69.45	84.94
BERT CLS-vector	78.68	84.85	94.21	88.23	84.13	91.4	71.13	84.66
InferSent - GloVe	81.57	86.54	92.50	**90.38**	84.18	88.2	75.77	85.59
Universal Sentence Encoder	80.09	85.19	93.98	86.70	86.38	**93.2**	70.14	85.10
SBERT-NLI-base	83.64	89.43	94.39	89.86	88.96	89.6	**76.00**	87.41
SBERT-NLI-large	**84.88**	**90.07**	**94.52**	90.33	**90.66**	87.4	75.94	**87.69**

Table 5: Evaluation of SBERT sentence embeddings using the SentEval toolkit. SentEval evaluates sentence embeddings on different sentence classification tasks by training a logistic regression classifier using the sentence embeddings as features. Scores are based on a 10-fold cross-validation.

https://github.com/UKPLab/ sentence-transformers

Show Your Work: Improved Reporting of Experimental Results





✓ **For all reported experimental results**

- ☐ Description of computing infrastructure
- ☐ Average runtime for each approach
- ☐ Details of train/validation/test splits
- ☐ Corresponding validation performance for each reported test result
- ☐ A link to implemented code

✓ **For experiments with hyperparameter search**

- ☐ Bounds for each hyperparameter
- ☐ Hyperparameter configurations for best-performing models
- ☐ Number of hyperparameter search trials
- ☐ The method of choosing hyperparameter values (e.g., uniform sampling, manual tuning, etc.) and the criterion used to select among them (e.g., accuracy)
- ☐ Expected validation performance, as introduced in §3.1, or another measure of the mean and variance as a function of the number of hyperparameter trials.

Text Box 1: Experimental results checklist.

https://github.com/allenai/ show-your-work

Thank you for you attention!