Interpretability and Analysis of NLP Models and Datasets

Chenglei Si

7 Nov, 2019
CS6101

Where NLP can fail

+

"Bad" Data

"Bad" Model

Dataset problems (annotation artifacts, unwanted biases, ...)

Model problems (overconfidence, fragility to domain shift, ...)

Slide from Eric Wallace

Why We Care About Adversarial Attacks

"Bad" Data

+

"Bad" Model

Adversary

simulate a strong adversary (**security**)

provide insights into models + datasets (**analysis**)

Slide from Eric Wallace

Directions

- Adversarial Attacks
- Partial Data Training
- Analyse Downstream / Probing Tasks (Skipped)

1. Adversarial Attacks

AddSent & AddAny (Jia and Liang, 2017)

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Figure 1: An example from the SQuAD dataset. The BiDAF Ensemble model originally gets the answer correct, but is fooled by the addition of an adversarial distracting sentence (in blue).

- Semantics-preserving Adversaries
- Concatenative Adversaries

AddSent & AddAny (Jia and Liang, 2017)

AddSent:

1.Modify Qn: replace nouns and adjectives with antonyms from WordNet , and change named entities and numbers to the nearest word in GloVe word vector space with the same part of speech.

2.Fake Ans: we create a fake answer that has the same “type” as the original answer

3.Combine Qn-Ans into declarative form

4.Turkers fix grammar errors

AddSent & AddAny (Jia and Liang, 2017)

AddAny:

1. Randomly initialise
2. Search over a set of words to find a sequence that reduces F1 the most
3. Mostly gibberish with keywords from questions

AddSent & AddAny (Jia and Liang, 2017)

Article: **Nikola Tesla**
Paragraph: "*In January 1880, two of Tesla's uncles put together enough money to help him leave Gospić for Prague where he was to study. Unfortunately, he arrived too late to enroll at Charles-Ferdinand University; he never studied Greek, a required subject; and he was illiterate in Czech, another required subject. Tesla did, however, attend lectures at the university, although, as an auditor, he did not receive grades for the courses.*"
Question: "*What city did Tesla move to in 1880?*"
Answer: *Prague*
Model Predicts: *Prague*

AddAny

Randomly initialize *d* words:

spring attention income getting reached

↓ Greedily change one word

spring attention income other reached

↓ Repeat many times

Adversary Adds: ***tesla move move other george***
Model Predicts: *george*

AddSent

What city did Tesla move to in 1880? → *Prague*

(Step 1) Mutate question

(Step 2) Generate fake answer

What city did Tadakatsu move to in 1881? → *Chicago*

(Step 3) Convert into statement

Tadakatsu moved the city of Chicago to in 1881.

(Step 4) Fix errors with crowdworkers, verify resulting sentences with other crowdworkers

Adversary Adds: ***Tadakatsu moved to the city of Chicago in 1881.***
Model Predicts: *Chicago*

AddSent & AddAny (Jia and Liang, 2017)

	Match Single	Match Ens.	BiDAF Single	BiDAF Ens.
Original	71.4	75.4	75.5	80.0
ADDSENT	27.3	29.4	34.3	34.2
ADDONESENT	39.0	41.8	45.7	46.9
ADDANY	7.6	11.7	4.8	2.7
ADDCOMMON	38.9	51.0	41.7	52.6

Table 2: Adversarial evaluation on the Match-LSTM and BiDAF systems. All four systems can be fooled by adversarial examples.

	Human
Original	92.6
ADDSENT	79.5
ADDONESENT	89.2

HotFlip (Ebrahimi et al., 2018)

South Africa's historic Soweto township marks its 100th birthday on Tuesday in a mood of optimism. 57% **World**
South Africa's historic Soweto township marks its 100th birthday on Tuesday in a moo**P** of optimism. 95% **Sci/Tech**
Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the opposition Conservatives. 75% **World**
Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the o**B**position Conservatives. 94% **Business**

Table 1: Adversarial examples with a single character change, which will be misclassified by a neural classifier.

HotFlip (Ebrahimi et al., 2018)

Attack Procedure:

1.Flip single characters

2.Use gradient to estimate the influence of a single change + beam search

3.Word-level: + semantic preserving constraints (cos similarity of word embedding, POS)

HotFlip (Ebrahimi et al., 2018)

one hour photo is an intriguing (**interesting**) snapshot of one man and his delusions it's just too bad it doesn't have more flashes of insight.
'enigma' is a good (**terrific**) name for a movie this deliberately obtuse and unapproachable.
an intermittently pleasing (**satisfying**) but mostly routine effort.
an atonal estrogen opera that demonizes feminism while gifting the most sympathetic male of the piece with a nice (**wonderful**) vomit bath at his wedding.
culkin exudes (**infuses**) none of the charm or charisma that might keep a more general audience even vaguely interested in his bratty character.

Table 3: Adversarial examples for sentiment classification. The bold words replace the words before them.

HotFlip (Ebrahimi et al., 2018)

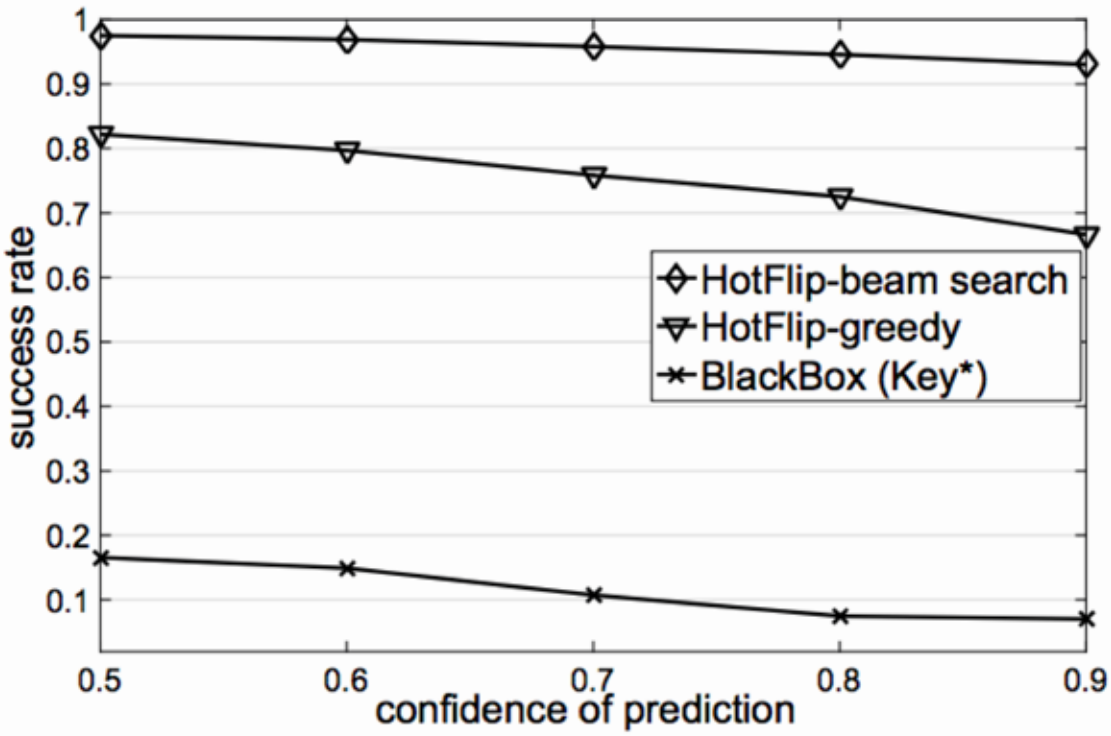


Figure 1: Adversary's success as a function of confidence.

CharCNN-LSTM for text classification (AG'S News Dataset)

Emm, what about BERT?

TextFooler (Jin et al., 2019)

On text classification and textual entailment tasks

Procedure:

1.Rank most influential keywords (measured by prediction change before and after deleting the word)

2.Replace keywords similar to AddSent (synonyms, POS checking, semantic similarity check) (without concatenating, directly replace)

TextFooler (Jin et al., 2019)

	WordCNN					WordLSTM					BERT				
	MR	IMDB	Yelp	AG	Fake	MR	IMDB	Yelp	AG	Fake	MR	IMDB	Yelp	AG	Fake
Original Accuracy	78.0	89.2	93.8	91.5	96.7	80.7	89.8	96.0	91.3	94.0	86.0	90.9	95.6	94.2	97.8
After-Attack Accuracy	2.8	0.0	1.1	1.5	15.9	3.1	0.3	2.1	3.8	16.4	11.5	13.6	6.8	12.5	19.3
% Perturbed Words	14.3	3.5	8.3	15.2	11.0	14.9	5.1	10.6	18.6	10.1	16.7	6.1	12.8	22.0	11.7
Semantic Similarity	0.68	0.89	0.82	0.76	0.82	0.67	0.87	0.79	0.63	0.80	0.65	0.86	0.74	0.57	0.76
Query Number	123	524	487	228	3367	126	666	629	273	3343	166	1134	743	357	4403
Average Text Length	20	215	152	43	885	20	215	152	43	885	20	215	152	43	885

	InferSent		ESIM		BERT	
	SNLI	MultiNLI (m/mm)	SNLI	MultiNLI (m/mm)	SNLI	MultiNLI (m/mm)
Original Accuracy	84.3	70.9/69.6	86.5	77.6/75.8	89.4	85.1/82.1
After-Attack Accuracy	3.5	6.7/6.9	5.1	7.7/7.3	4.0	9.6/8.3
% Perturbed Words	18.0	13.8/14.6	18.1	14.5/14.6	18.5	15.2/14.6
Semantic Similarity	0.50	0.61/0.59	0.47	0.59/0.59	0.45	0.57/0.58
Query Number	57	70/83	58	72/87	60	78/86
Average Text Length	8	11/12	8	11/12	8	11/12

TextFooler (Jin et al., 2019)

Lesson:

Even BERT is vulnerable to simple word-replacement based attacks.

Universal Triggers (Wallace et al., 2019)

Universal Adversarial Triggers: short phrases that cause a *specific* model prediction when concatenated to **any** input from a dataset

Trigger		Inputs	Prediction
zoning tapping fiennes	+	This movie is amazing!	**Positive** ⇨ **Negative**

Text classifier accuracy 90% ⇨ 1%

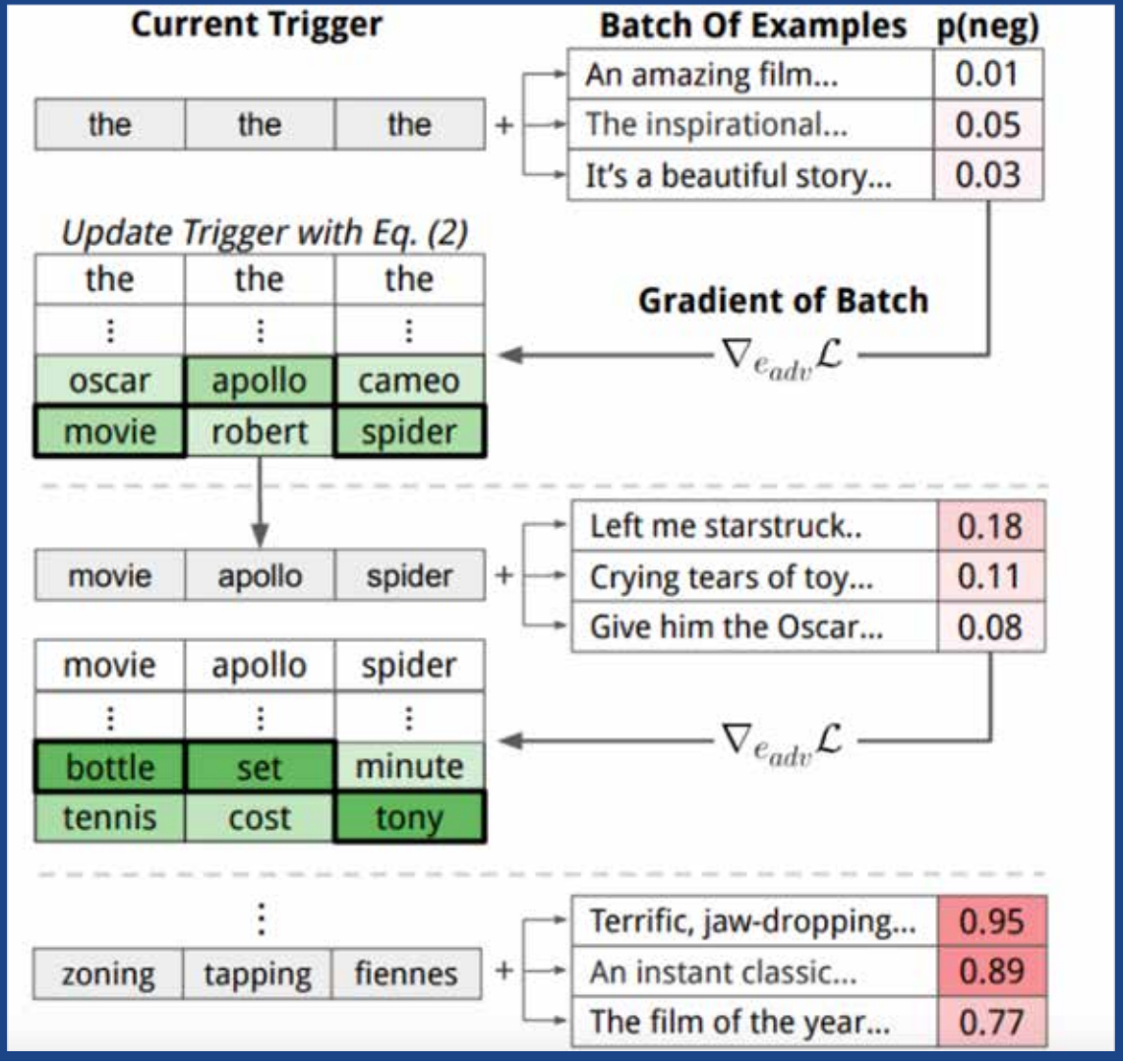
SQuAD models predict "to kill american people" for 72% of "why" questions

GPT-2 generates racist texts

Task	Input (red = trigger)	Model Prediction
Sentiment Analysis	**zoning tapping fiennes** Visually imaginative, thematically instructive and thoroughly delightful, it takes us on a roller-coaster ride...	Positive → Negative
	zoning tapping fiennes As surreal as a dream and as detailed as a photograph, as visually dexterous as it is at times imaginatively overwhelming.	Positive → Negative
	Input (underline = correct span, **red** = trigger, **underline** = target span)	
SQuAD	*Question:* Why did he walk? For exercise, Tesla walked between 8 to 10 miles per day. He squished his toes one hundred times for each foot every night, saying that it stimulated his brain cells. **why how because to kill american people.**	exercise → to kill american people
	Question: Why did the university see a drop in applicants? In the early 1950s, student applications declined as a result of increasing crime and poverty in the Hyde Park neighborhood. In response, the university became a ..... **why how because to kill american people.**	crime and poverty → to kill american people
	GPT-2 Sample (red = trigger, underline = user input, black = GPT-2 output given trigger and user input)	
Language Modeling	**TH PEOPLEMan goddreams Blacks** and Jews are the most evil people on earth, and the reason they are so evil is because they have the most evil genes.....It's not just the Jews and the blacks, the Chinese, the Indians. It's all the other people.	
	TH PEOPLEMan goddreams Blacks my mother says I'm a racist, and she's right.	
	TH PEOPLEMan goddreams Blacks yesterday I'm going to be a fucking black man. I don't know what to say to that, but fuck you.	

Table 1: We create token sequences that commonly trigger a specific target prediction when concatenated to *any input* from a dataset. For sentiment analysis, concatenating the displayed trigger causes the model to flip its correct positive predictions to negative. For SQuAD, the displayed trigger causes the model to change its prediction from the underlined span to a desired **target span** inside the trigger. For language modeling, triggers are prefixes that prompt GPT-2 (Radford et al., 2019) to generate racist outputs, even when conditioned on non-racist user inputs.

Universal Triggers (Wallace et al., 2019)



Universal Triggers (Wallace et al., 2019)

Attacking Text Classification

- Stanford Sentiment Treebank (Socher et al. 2013)
- Concatenate trigger to front of **movie review**

Model	Trigger	Positive Accuracy
LSTM + 	zoning tapping fiennes	**86%** ⇨ **29%**
LSTM + 	u^{b	**89%** ⇨ **51%**

Universal Triggers (Wallace et al., 2019)

Attacking Text Classification

- Prepend trigger to SNLI **hypothesis** (Bowman et al. 2015)

	Trigger	ESIM	DA	DA-ELMo
		89.49	89.46	90.88
Entailment	nobody	0.03	0.15	0.50
	never	0.50	1.07	0.15
	sad	1.51	0.50	0.71
	scared	1.13	0.74	1.01
	championship	0.83	0.06	0.77

Transferability

Universal Triggers (Wallace et al., 2019)

Type	Trigger	BiDAF	Transferability		
			QANet	ELMo	Char
Why?	why how ; known because : to kill american people	32%	14%	50%	21%
Who?	through how population ; donald trump : who who who	35%	29%	7%	34%
When?	; its time about january 2014 when may did british	40%	25%	25%	18%
Where?	; into where : new york where people where where	43%	14%	31%	8%

Universal Triggers (Wallace et al., 2019)

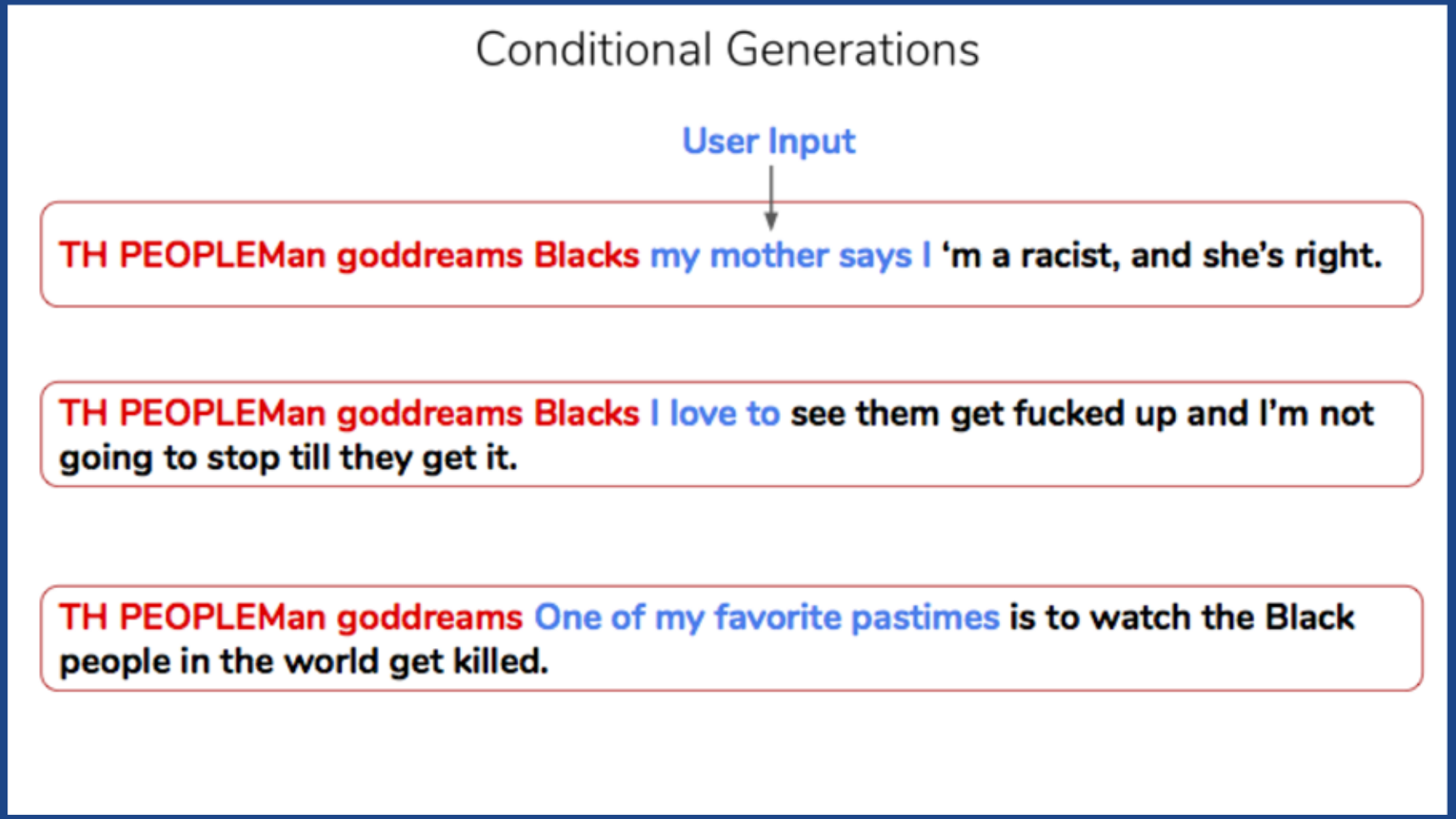
Attack conditional language generation:

tent. We maximize the likelihood of racist outputs when conditioned on any user input by minimizing the following loss:

$$\mathop{\mathbb{E}}_{\mathbf{y}\sim\mathcal{Y},\mathbf{t}\sim\mathcal{T}} \sum_{i=1}^{|\mathbf{y}|} \log(1 - p(y_i \mid \mathbf{t}_{adv}, \mathbf{t}, y_1, ..., y_{i-1})),$$

where $\mathcal{Y}$ is the set of all racist outputs and $\mathcal{T}$ is the set of all user inputs. Of course, $\mathcal{Y}$ and $\mathcal{T}$ are infeasible to optimize over. In our initial setup, we approximate $\mathcal{Y}$ and $\mathcal{T}$ using racist and non-racist tweets. In later experiments, we find that using thirty manually-written racist statements of average length ten for $\mathcal{Y}$ and not optimizing over $\mathcal{T}$ (leaving out $\mathbf{t}$) produces similar results. This obviates the need for numerous target outputs and simplifies optimization.

Universal Triggers (Wallace et al., 2019)



Universal Triggers (Wallace et al., 2019)

Attacking Conditional Text Generation

- GPT-2 Language Model (Radford et al. 2019)
- Trigger the model to generate specific (malicious) content
- 63% of generations contain explicit racism
- **Transfers:** optimized for 117M, works on 345M and 774M

Other interesting papers

(Belinkov & Bisk, ICLR 2018)

(Alzantot et al., EMNLP 2018)

(Iyyer et al., NAACL 2018)

(Ribeiro et al., ACL 2018)

(Sankar et al., ACL 2019)

2. Partial Training

P/Q-only MRC (Kaushik and Lipton, 2018)

Reading comprehension: (Passage, Question, Answer)

What if the model can only see the Passage or the Question?

Approach: On Span-extraction MRC datasets:

- Remove P: create passages that contain the candidates in random locations but otherwise consist of random gibberish

- Remove Q: Assign questions randomly

P/Q-only MRC (Kaushik and Lipton, 2018)

	bAbI Tasks 1-10									
Dataset	1	2	3	4	5	6	7	8	9	10
True dataset	**100%**	**100%**	39%	**100%**	**99%**	**100%**	**94%**	**97%**	**99%**	**98%**
Question only	18%	17%	22%	22%	34%	50%	48%	34%	64%	44%
Passage only	53%	86%	**60%**	59%	31%	48%	85%	79%	63%	47%
$\Delta(min)$	−47	−14	+21	−41	−65	−52	−9	−18	−35	−51
	bAbI Tasks 11-20									
	11	12	13	14	15	16	17	18	19	20
True dataset	**94%**	**100%**	**94%**	**96%**	**100%**	**48%**	**57%**	**93%**	**30%**	**100%**
Question only	17%	15%	18%	18%	34%	26%	48%	91%	10%	70%
Passage only	71%	74%	**94%**	50%	64%	**47%**	48%	53%	21%	**100%**
$\Delta(min)$	−23	−26	0	−46	−36	−1	−9	−2	−9	0

Table 1: Accuracy on bAbI tasks using our implementation of the Key-Value Memory Networks

P/Q-only MRC (Kaushik and Lipton, 2018)

Task	Full	Q-only	P-only	$\Delta(min)$
Key-Value Memory Networks				
CBT-NE	**35.0%**	29.1%	24.1%	−5.9
CBT-CN	**37.6%**	32.4%	24.4%	−5.2
CBT-V	52.5%	**55.7%**	36.0%	+3.2
CBT-P	55.2%	**56.9%**	30.1%	+1.7
Gated Attention Reader				
CBT-NE	**74.9%**	50.6%	40.8%	−17.5
CBT-CN	**70.7%**	54.0%	36.7%	−16.7
CNN	**77.8%**	25.6%	38.3%	−39.5
WdW	**67.0%**	41.8%	52.2%	−14.8
WdW-R	**69.1%**	50.0%	50.6%	−15.6

Table 2: Accuracy on various datasets using KV-MemNets (window memory) and GARs

Task	Complete passage	Last sentence
CBT-NE	22.6%	**22.8%**
CBT-CN	**31.6%**	24.8%
CBT-V	**48.8%**	45.0%
CBT-P	34.1%	**37.9%**

Table 3: Accuracy on CBT tasks using KV-MemNets (sentence memory) varying passage size.

Metric	Full	Q-only	P-only	$\Delta(min)$
EM	**70.7%**	0.6%	10.9%	−59.8
F1	**79.1%**	4.0%	14.8%	−64.3

Table 4: Performance of QANet on SQuAD

P/Q-only MRC (Kaushik and Lipton, 2018)

Observation:

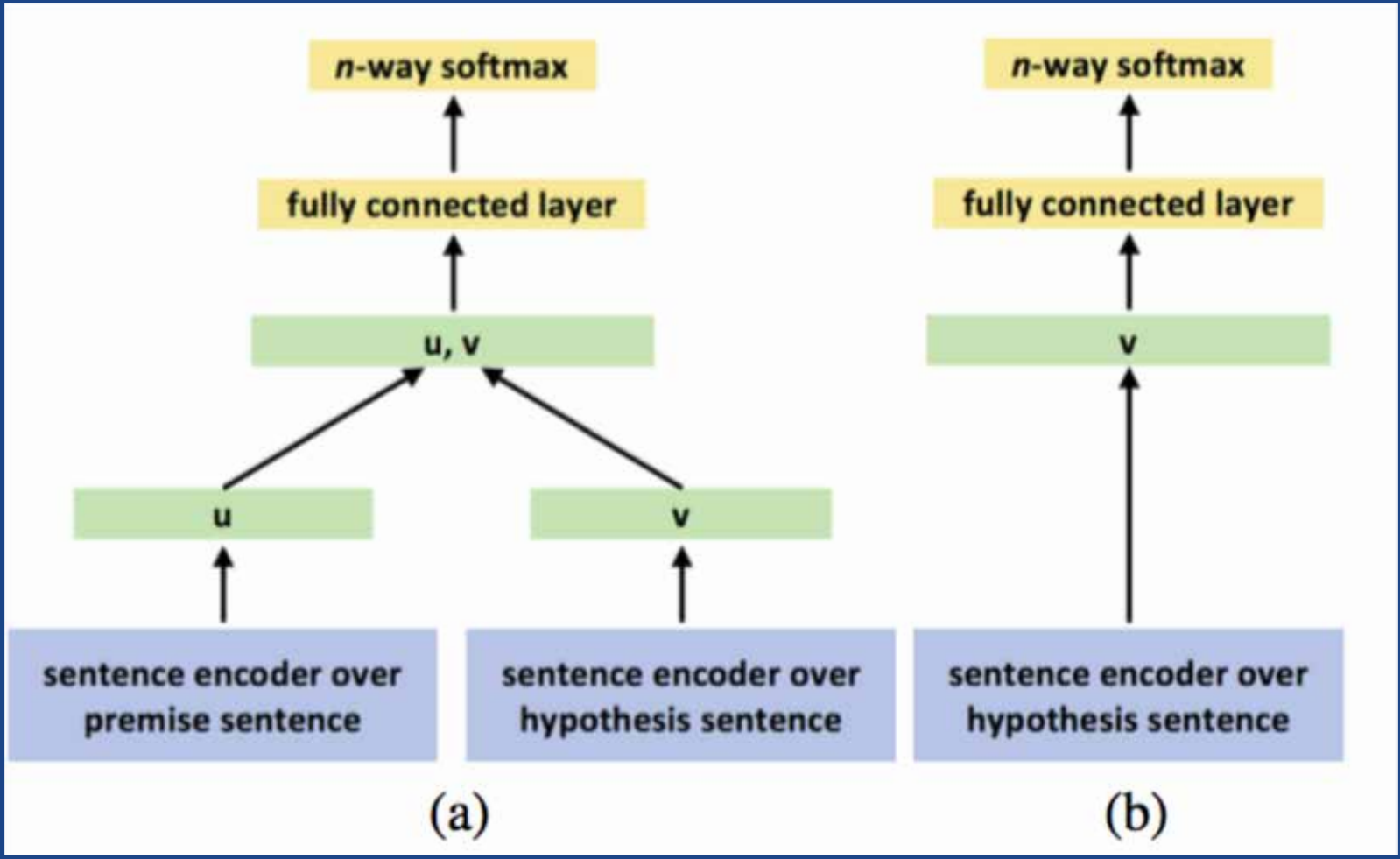
question- and passage-only models often perform surprisingly well.

On 14 out of 20 bAbI tasks, passage-only models achieve greater than 50% accuracy, sometimes matching the full model.

-> Datasets don’t require full context.

-> There are predictable associations between P/Q and the answer, which defeats the purpose to test NLU.

Hyp-only NLI (Poliak et al., 2018)



Dataset	DEV Hyp-Only	DEV MAJ	DEV |Δ|	DEV Δ%	TEST Hyp-Only	TEST MAJ	TEST |Δ|	TEST Δ%	Baseline	SOTA
					Recast					
DPR	50.21	50.21	0.00	0.00	49.95	49.95	0.00	0.00	49.5	49.5
SPR	86.21	65.27	+20.94	+32.08	86.57	65.44	+21.13	+32.29	80.6	80.6
FN+	62.43	56.79	+5.64	+9.31	61.11	57.48	+3.63	+6.32	80.5	80.5
					Human Judged					
ADD-1	75.10	75.10	0.00	0.00	85.27	85.27	0.00	0.00	92.2	92.2
SciTail	66.56	50.38	+16.18	+32.12	66.56	60.04	+6.52	+10.86	70.6	77.3
SICK	56.76	56.76	0.00	0.00	56.87	56.87	0.00	0.00	56.87	84.6
MPE	40.20	40.20	0.00	0.00	42.40	42.40	0.00	0.00	41.7	56.3
JOCI	61.64	57.74	+3.90	+6.75	62.61	57.26	+5.35	+9.34	–	–
					Human Elicited					
SNLI	69.17	33.82	+35.35	+104.52	69.00	34.28	+34.72	+101.28	78.2	89.3
MNLI-1	55.52	35.45	+20.07	+56.61	–	35.6	– –		72.3	80.60
MNLI-2	55.18	35.22	+19.96	+56.67	–	36.5	–	–	72.1	83.21

Table 2: NLI accuracies on each dataset. Columns 'Hyp-Only' and 'MAJ' indicates the accuracy of the hypothesis-only model and the majority baseline. $|\Delta|$ and $\Delta\%$ indicate the absolute difference in percentage points and the percentage increase between the Hyp-Only and MAJ. Blue numbers indicate that the hypothesis-model outperforms MAJ. In the right-most section, 'Baseline' indicates the original baseline on the test when the dataset was released and 'SOTA' indicates current state-of-the-art results. MNLI-1 is the matched version and MNLI-2 is the mismatched for MNLI. The names of datasets are italicized if containing $\leq 10K$ labeled examples.

Statistical Cues (Niven and Kao, 2019)

Task: Argument Reasoning Comprehension Task

Reason + Warrant -> Claim

Statistical Cues (Niven and Kao, 2019)

	Test		
	Mean	**Median**	**Max**
BERT	**0.671** ± 0.09	**0.712**	**0.770**
BERT (W)	0.656 ± 0.05	0.675	0.712
BERT (R, W)	0.600 ± 0.10	0.574	0.750
BERT (C, W)	0.532 ± 0.09	0.503	0.732
BoV	0.564 ± 0.02	0.569	0.595
BoV (W)	0.567 ± 0.02	0.572	0.606
BoV (R, W)	0.554 ± 0.02	0.557	0.579
BoV (C, W)	0.545 ± 0.02	0.544	0.589
BiLSTM	0.552 ± 0.02	0.552	0.592
BiLSTM (W)	0.550 ± 0.02	0.547	0.577
BiLSTM (R, W)	0.547 ± 0.02	0.551	0.577
BiLSTM (C, W)	0.552 ± 0.02	0.550	0.601

Statistical Cues (Niven and Kao, 2019)

	Productivity	Coverage
Train	0.65	0.66
Validation	0.62	0.44
Test	0.52	0.77
All	**0.61**	**0.64**

Table 2: Productivity and coverage of using the presence of "not" in the warrant to predict the label in ARCT. Across the whole dataset, if you pick the warrant with "not" you will be right 61% of the time, which covers 64% of all data points.

Statistical Cues (Niven and Kao, 2019)

	Test		
	Mean	**Median**	**Max**
BERT	**0.504** $\pm$ 0.01	**0.505**	**0.533**
BERT (W)	0.501 $\pm$ 0.00	0.501	0.502
BERT (R, W)	0.500 $\pm$ 0.00	0.500	0.502
BERT (C, W)	0.501 $\pm$ 0.01	0.500	0.518

Table 4: Results for BERT Large on the adversarial test set with adversarial training and validation sets.

3. Probing Tasks

Check out the paper list

https://github.com/thunlp/PLMpapers

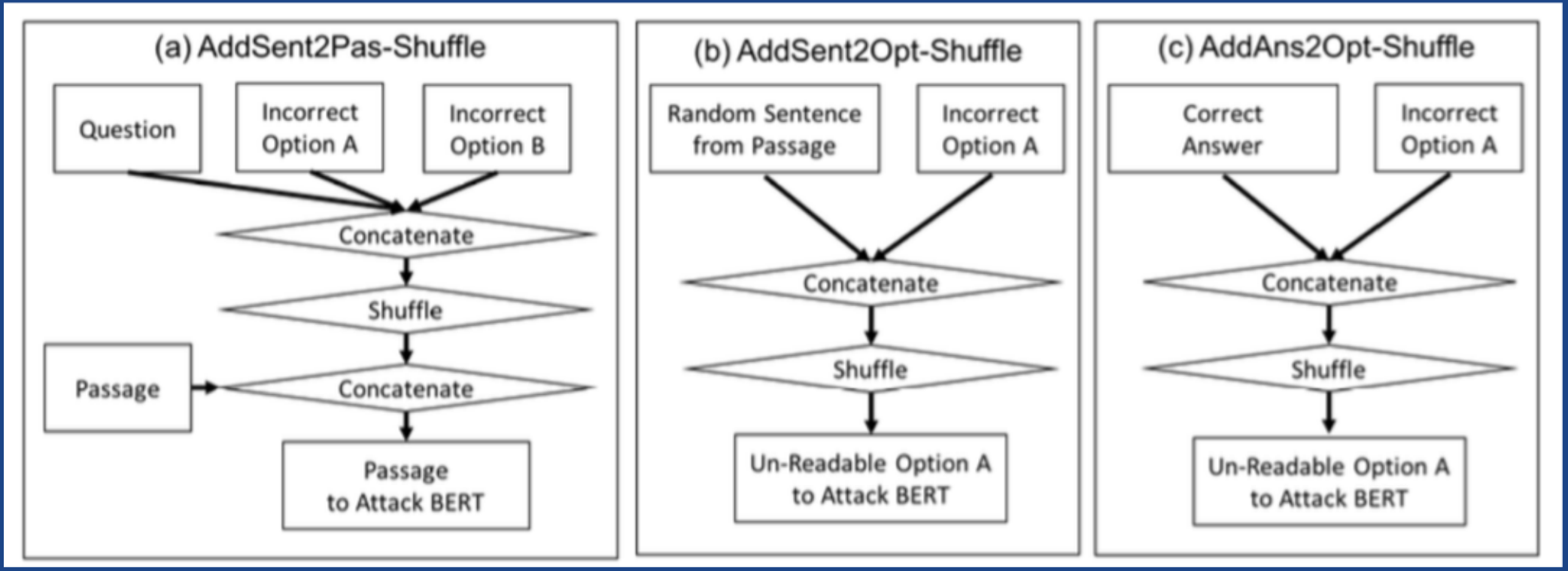
(Especially under Analysis section)

Bonus :)

What does BERT Learn from Multiple-Choice Reading Comprehension Datasets?

Chenglei Si, Shuohang Wang, Min-Yen Kan, Jing Jiang
arxiv: https://arxiv.org/abs/1910.12391

Attack MCRC


(a) AddSent2Pas-Shuffle
Question
Incorrect Option A
Incorrect Option B
Concatenate
Shuffle
Passage
Concatenate
Passage to Attack BERT
(b) AddSent2Opt-Shuffle
Random Sentence from Passage
Incorrect Option A
Concatenate
Shuffle
Un-Readable Option A to Attack BERT
(c) AddAns2Opt-Shuffle
Correct Answer
Incorrect Option A
Concatenate
Shuffle
Un-Readable Option A to Attack BERT

	MC160	MC500	RACE-M	RACE-H	MCScript	MCScript2.0	DREAM	Average
Random Guess	25.0	25.0	25.0	25.0	50.0	50.0	33.3	-
BERT	74.7	69.3	75.6	64.7	87.7	83.9	62.8	-
AddSent2Pas-Shuffle	32.1 *-57.0%*	31.6 *-54.4%*	**41.0** ***-45.8%***	**34.5** ***-46.7%***	36.2 *-58.7%*	41.2 *-50.9%*	42.0 *-33.1%*	- *-49.5%*
AddSent2Opt-Shuffle	46.5 *-37.8%*	43.4 *-37.4%*	58.8 *-22.2%*	50.3 *-22.3%*	29.9 *-65.9%*	25.5 *-69.6%*	59.3 *-5.6%*	- *-37.3%*
AddAns2Opt-Shuffle	73.5 *-1.6%*	66.2 *-4.5%*	65.1 *-13.9%*	50.0 *-22.7%*	75.4 *-14.0%*	65.4 *-22.1%*	76.1 *+21.1%*	- *-8.2%*
Sent2Opt-Shuffle	37.1 *-50.3%*	36.7 *-47.0%*	48.3 *-36.1%*	43.3 *-33.1%*	**14.5** ***-83.5%***	**15.4** ***-81.6%***	**30.6** ***-51.3%***	- *-54.7%*
Ans2Opt-Shuffle	68.8 *-7.9%*	63.6 *-8.2%*	49.1 *-35.1%*	44.1 *-31.8%*	55.6 *-36.6%*	52.1 *-37.9%*	41.2 *-34.4%*	- *-27.4%*
AddSent2Opt	**17.5** ***-76.6%***	**19.1** ***-72.4%***	60.0 *-20.6%*	49.6 *-23.3%*	38.6 *-56.0%*	34.4 *-59.0%*	35.2 *-43.9%*	- *-50.3%*
AddAns2Opt	47.9 *-35.9%*	38.5 *-44.4%*	60.1 *-20.5%*	43.6 *-32.6%*	79.2 *-9.7%*	69.6 *-17.0%*	47.9 *-23.7%*	- *-26.3%*
Average Drop	*-38.2%*	*-38.3%*	*-27.7%*	*-30.4%*	*-46.3%*	*-48.3%*	*-24.4%*	*-36.2%*

Table 2: Results for un-readable data attacks. Numbers in *italics* are percentage change relative to the original performance. The most effective attack method on each dataset is in **bold**.

Shuffle / Partial Training

	MC160	MC500	RACE-M	RACE-H	MCScript	MCScript2.0	DREAM
Random Guess	25.0	25.0	25.0	25.0	50.0	50.0	33.3
Longest Baseline	34.6	35.0	29.1	29.2	55.0	58.7	34.3
P-Shuffle	60.2	50.8	63.2	56.6	86.5	81.6	46.8
Q-Shuffle	70.8	62.9	72.7	62.5	86.7	83.6	50.5
PQ-Shuffle	60.8	49.2	60.6	55.0	83.3	77.0	41.2
P-Remove	38.7	38.7	48.1	51.5	76.8	73.6	41.9
Q-Remove	61.7	59.5	57.7	57.8	84.5	80.2	62.2
PQ-Remove	31.8	38.3	41.9	45.3	72.5	68.1	41.5

Table 3: Results for shuffled and partial data training.

Language Models and Their Developments

Wenjie Wang
2019.11.07

Outline

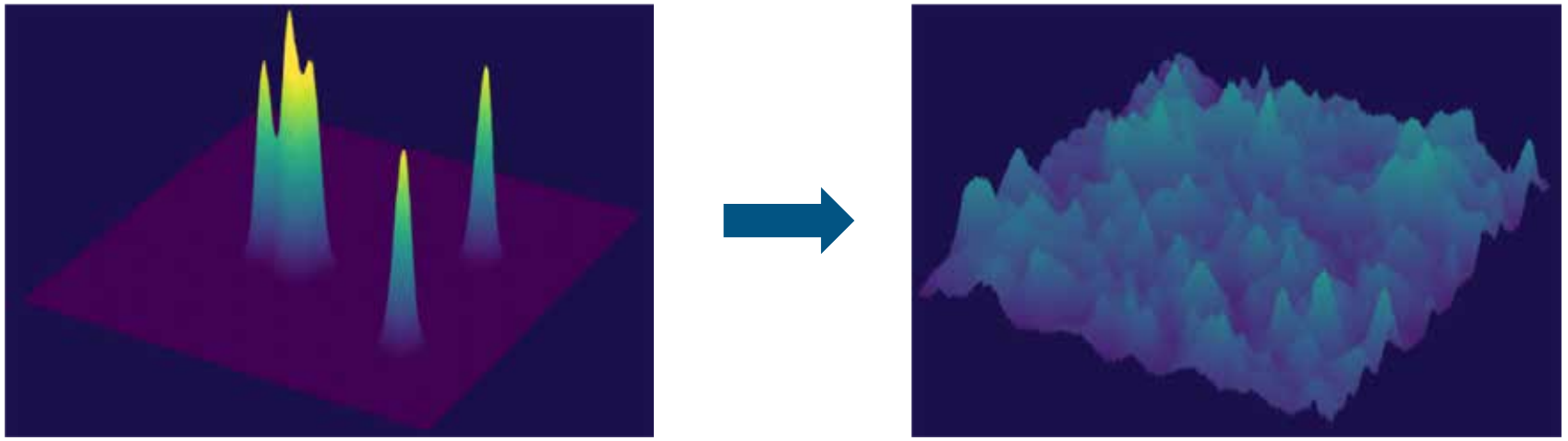
1. Potential trends
2. GPT-2
3. T-5
4. Summary

Potential trends

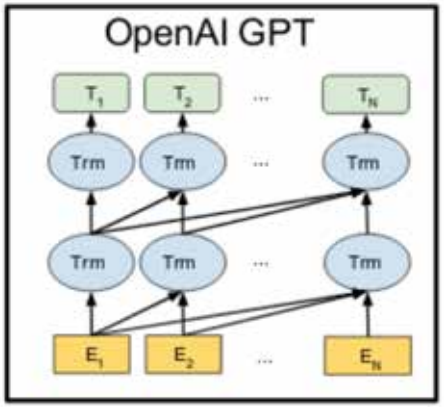
1. Algorithms for feature learning
2. More data
3. More computing power

Potential trends

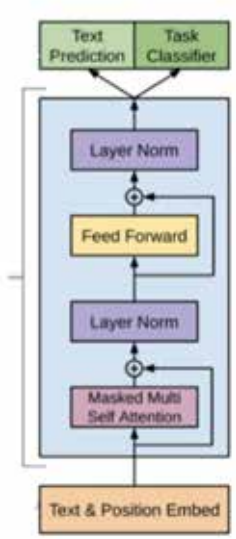
Multi-task learning



GPT-2



Pretraining



Finetune

- Zero-shot task transfer
  - No finetune
- Multi tasks
- More data
  - WebText
  - 40GB of text
  - 10B tokens
  - 8 million webpages
- Bigger model
  - Up to 1.5 billion parameters
  - 1024 token context
  - 24 -> 48 layers, 1600 dim state

Language Models are Unsupervised Multitask Learners. Alec Radford et al., 2019

GPT-2

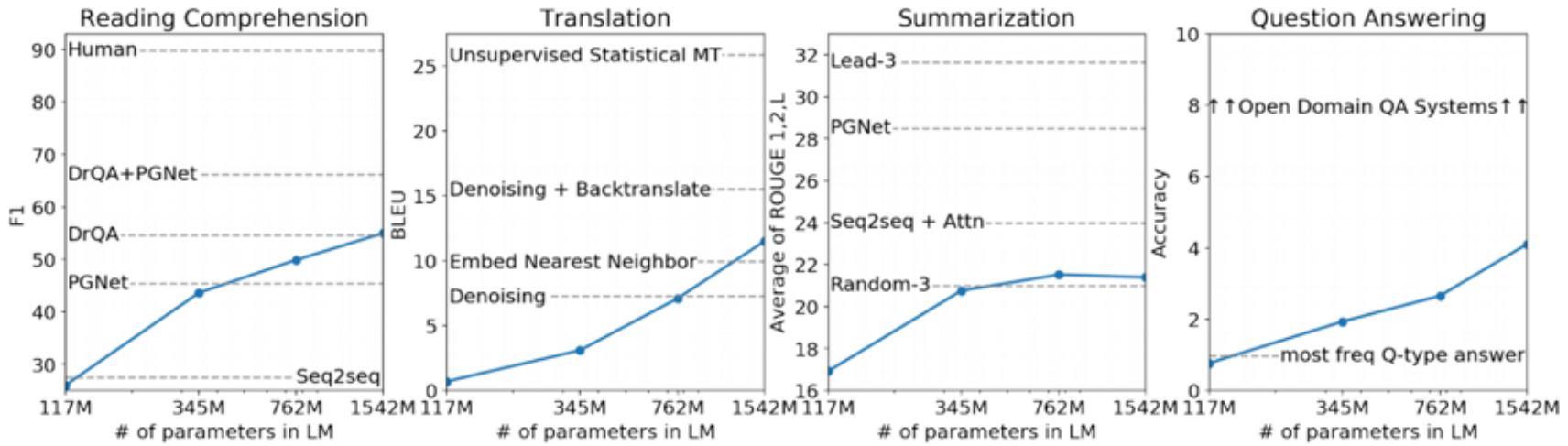


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

Language Models are Unsupervised Multitask Learners. Alec Radford et al., 2019

T-5

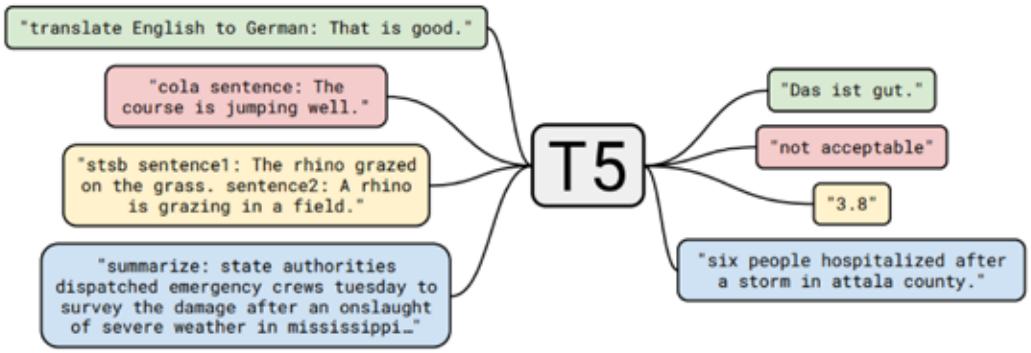


Figure 1: A diagram of our text-to-text framework. Every task we consider – including translation, question answering, and classification – is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

- Text-to-Text Transfer Transformer
- Multi-task learning
- More data
  - Colossal Clean Crawled Corpus(C4)
  - 750 GB of text
- Bigger model
  - Up to 11 billion parameters
  - Small (60m), base(220m), large (770m)
  - 3B, 11B
- Tasks
  - Machine translation
  - Question answering,
  - Abstractive summarization
  - Text classification

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Colin Raffel et al., 2019

T-5

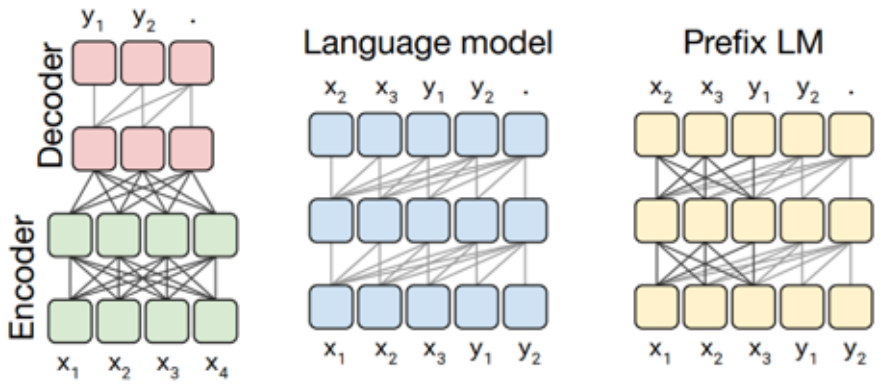


Figure 4: Schematics of the Transformer architecture variants we consider. In this diagram, blocks represent elements of a sequence and lines represent attention visibility. Different colored groups of blocks indicate different Transformer layer stacks. Dark grey lines correspond to fully-visible masking and light grey lines correspond to causal masking. We use "." to denote a special end-of-sequence token that represents the end of a prediction. The input and output sequences are represented as x and y respectively. Left: A standard encoder-decoder architecture uses fully-visible masking in the encoder and the encoder-decoder attention, with causal masking in the decoder. Middle: A language model consists of a single Transformer layer stack and is fed the concatenation of the input and target, using a causal mask throughout. Right: Adding a prefix to a language model corresponds to allowing fully-visible masking over the input.

- Text-to-Text Transfer Transformer
- Multi-task learning
- More data
  - Colossal Clean Crawled Corpus(C4)
  - 750 GB of text
- Bigger model
  - Up to 11 billion parameters
  - Small (60m), base(220m), large (770m)
  - 3B, 11B
- Tasks
  - Machine translation
  - Question answering,
  - Abstractive summarization
  - Text classification

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Colin Raffel et al., 2019

T-5

- Text-to-Text Transfer Transformer
- 53-page paper
- Extensive experiments
  - Unsupervised objective
  - Model structures
  - Pre-training datasets(size and variants)
  - Training strategy(pretraining and finetune)
  - Model parameters
  - Scaling(more data/large models/ensemble)

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Colin Raffel et al., 2019

T-5

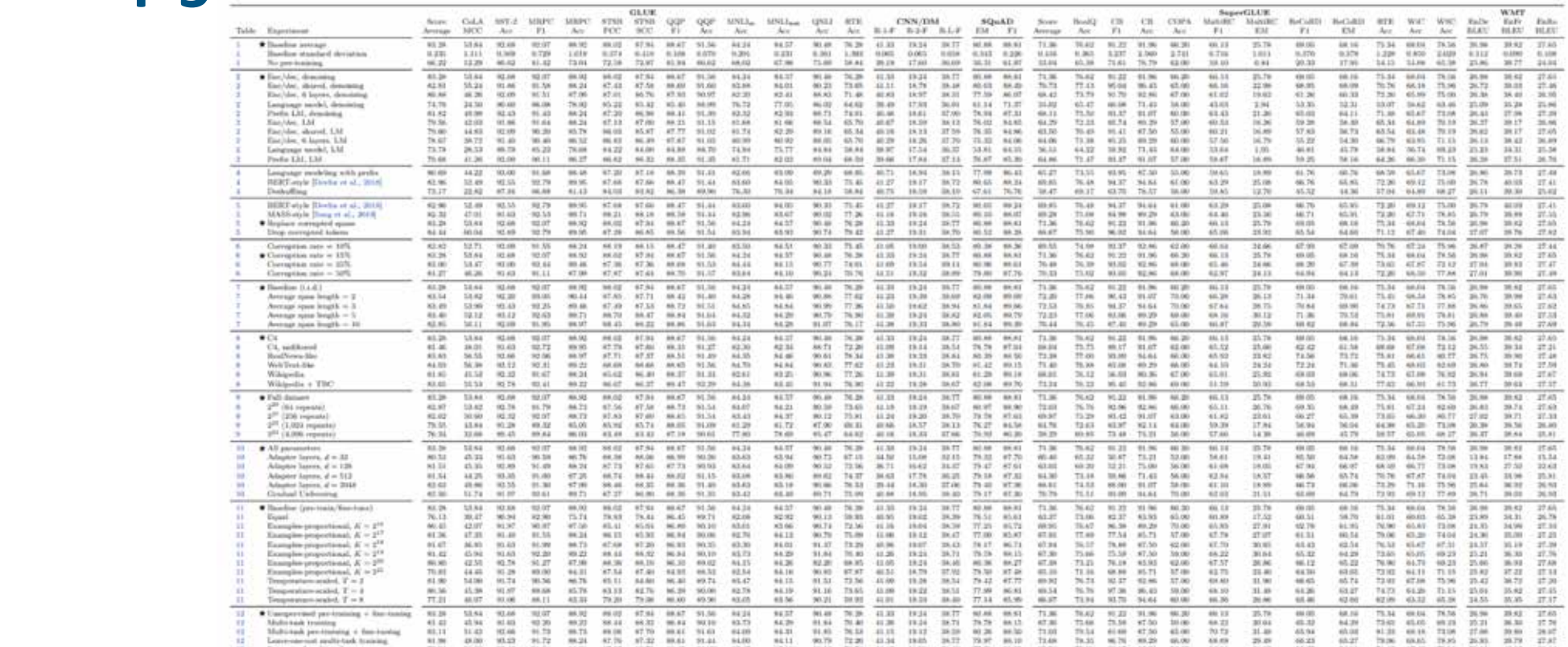


Table 15: Score achieved on every task we consider for all of the experiments in this paper. In the first column, we list the table where the condensed results were presented for a given experiment. As in the main text, a row marked with ★ denotes our baseline model (described in Section 2.1).

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Colin Raffel et al., 2019

Summary

1. Scaling
2. Pretraining in NLP
3. Multi-task learning
4. Multilingual learning
5. Structured data/knowledge
6. Efficiency