Using the Transformer

T5 (Raffel et al., 2019)

Learning goals

- Understand the improvements over BERT
- Dynamic Masking

GOOGLE'S T5 ‣ RAFFEL ET AL. (2019)

Short summary:

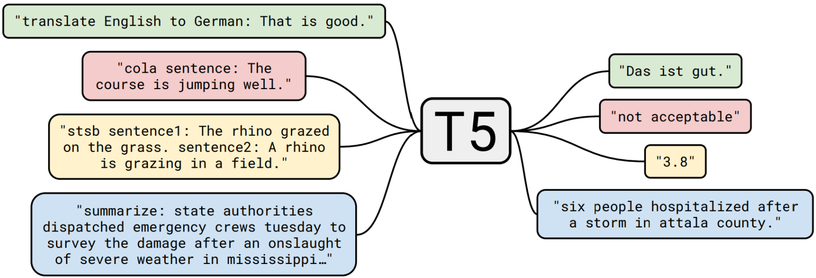
- *Text-to-Text Transfer Transformer (T5)*
- A complete encoder-decoder Transformer architecture
- Relative positional emeddings
- All tasks reformulated as text-to-text tasks

$\rightarrow$ Probably the most important innovation of this work

- From BERT-size up to 11 Billion parameters
- Paradigm shift from *Sequential Transfer Learning* towards *Multi-Task Learning*

‣ Animation (Link)

ILLUSTRATION



Source: Raffel et al. (2019)

INPUT AND OUTPUT FORMAT

Input Side:

- SentencePiece framework w/ WordPiece tokens
- Add task-specific (text) prefix to the original input
- Choice of the prefix: Hyperparameter!!

$\rightarrow$ Changing this had limited impact

$\rightarrow$ No extensive experiments performed by the authors

Output Side:

- Output as a word or a piece of text (also similarity scores)
- If output not present in set of potential alternatives, prediction considered as wrong

ADD-ON: DISTINCTION TO PROMPTING

Adding task-specific (text) prefix:

- Add task-specific (text) prefix to the original input
- Model is fine-tuned on samples prepended with this prefix

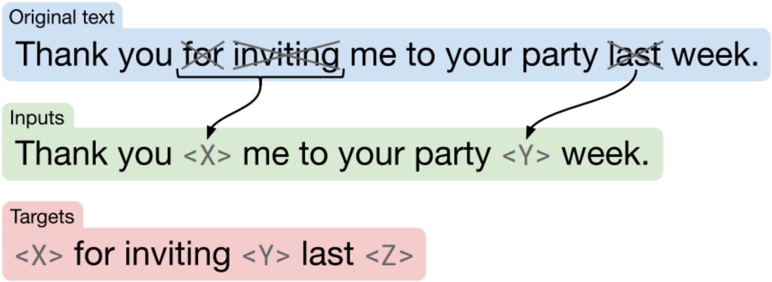
$\rightarrow$ It learns to recognize what it is expected to do when encountering a specific prefix at test time

$\rightarrow$ Probably because of this: limited impact found by the authors

Prompting:

- Refers to just querying a trained w/o fine-tuning it (cf. next chapter)
- Paradigm of Few-/Zero-Shot Learning
- This is found to have a *huge* impact on model performance

PRE-TRAINING T5



Baseline objective (Source: Raffel et al., 2019)

1. Mark spans in input sequence for corruptions
2. Replace tokens with sentinel tokens
3. Predict sentinel tokens and replaced tokens

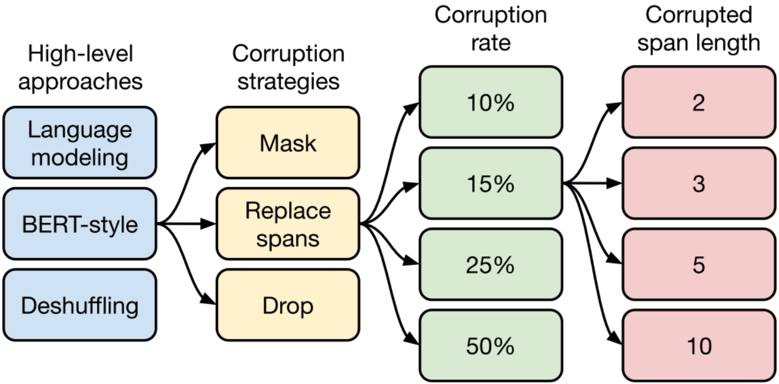
PRE-TRAINING OBJECTIVES

- Authors experimented with different objectives
- Most of them rely in some way on MLM

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

Source: Raffel et al. (2019)

PRE-TRAINING OBJECTIVES



Source: Raffel et al. (2019)

THE COLOSSAL CLEAN CRAWLED CORPUS (C4)

- Effort to measure the effect of quality, characteristics & size of the pre-training resources
- Common Crawl as basis, careful cleaning and filtering for English language
- Orders of magnitude larger (750GB) compared to commonly used corpora

THE COLOSSAL CLEAN CRAWLED CORPUS (C4)

Experiments (with respect to C4)

Data set	Size	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ C4	745GB	83.28	**19.24**	80.88	71.36	**26.98**	**39.82**	**27.65**
C4, unfiltered	6.1TB	81.46	19.14	78.78	68.04	26.55	39.34	27.21
RealNews-like	35GB	**83.83**	**19.23**	80.39	72.38	**26.75**	**39.90**	**27.48**
WebText-like	17GB	**84.03**	**19.31**	**81.42**	71.40	**26.80**	**39.74**	**27.59**
Wikipedia	16GB	81.85	**19.31**	81.29	68.01	**26.94**	39.69	**27.67**
Wikipedia + TBC	20GB	83.65	**19.28**	**82.08**	**73.24**	**26.77**	39.63	**27.57**

Number of tokens	Repeats	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Full data set	0	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	**39.82**	**27.65**
2^{29}	64	**82.87**	**19.19**	**80.97**	**72.03**	**26.83**	**39.74**	**27.63**
2^{27}	256	82.62	**19.20**	79.78	69.97	**27.02**	**39.71**	27.33
2^{25}	1,024	79.55	18.57	76.27	64.76	26.38	39.56	26.80
2^{23}	4,096	76.34	18.33	70.92	59.29	26.37	38.84	25.81

Source: Raffel et al. (2019)

T5 - EXHAUSTIVE EXPERIMENTS

Performed experiments with respect to ..

- *.. architecture, size & objective*

$\rightarrow$ enc, dec, enc-dec

$\rightarrow$ Between 60M and 11B parameters

- *.. details of the Denoising objective (which was found to work best)*
- *.. fine-tuning methods*

$\rightarrow$ Adapter layers

$\rightarrow$ Gradual Unfreezing (cf. ULMFiT)

- *.. Multi-task learning strategies*

$\rightarrow$ Examples-proportional mixing

$\rightarrow$ Temperature-scaled mixing

BENCHMARK RESULTS

Model	GLUE Average	CoLA Matthew's	SST-2 Accuracy	MRPC F1	MRPC Accuracy	STS-B Pearson	STS-B Spearman
Previous best	89.4[a]	69.2[b]	97.1[a]	**93.6**[b]	**91.5**[b]	92.7[b]	92.3[b]
T5-Small	77.4	41.0	91.8	89.7	86.6	85.6	85.0
T5-Base	82.7	51.1	95.2	90.7	87.5	89.4	88.6
T5-Large	86.4	61.2	96.3	92.4	89.9	89.9	89.2
T5-3B	88.5	67.1	97.4	92.5	90.0	90.6	89.8
T5-11B	**90.3**	**71.6**	**97.5**	92.8	90.4	**93.1**	**92.8**

Model	QQP F1	QQP Accuracy	MNLI-m Accuracy	MNLI-mm Accuracy	QNLI Accuracy	RTE Accuracy	WNLI Accuracy
Previous best	74.8[c]	**90.7**[b]	91.3[a]	91.0[a]	**99.2**[a]	89.2[a]	91.8[a]
T5-Small	70.0	88.0	82.4	82.3	90.3	69.9	69.2
T5-Base	72.6	89.4	87.1	86.2	93.7	80.1	78.8
T5-Large	73.9	89.9	89.9	89.6	94.8	87.2	85.6
T5-3B	74.4	89.7	91.4	91.2	96.3	91.1	89.7
T5-11B	**75.1**	90.6	**92.2**	**91.9**	96.9	**92.8**	**94.5**

Results on GLUE (Source: Raffel et al., 2019)

BENCHMARK RESULTS

Model	SQuAD EM	SQuAD F1	SuperGLUE Average	BoolQ Accuracy	CB F1	CB Accuracy	COPA Accuracy
Previous best	90.1[a]	95.5[a]	84.6[d]	87.1[d]	90.5[d]	95.2[d]	90.6[d]
T5-Small	79.10	87.24	63.3	76.4	56.9	81.6	46.0
T5-Base	85.44	92.08	76.2	81.4	86.2	94.0	71.2
T5-Large	86.66	93.79	82.3	85.4	91.6	94.8	83.4
T5-3B	88.53	94.95	86.4	89.9	90.3	94.4	92.0
T5-11B	**91.26**	**96.22**	**88.9**	**91.2**	**93.9**	**96.8**	**94.8**
Model	**MultiRC F1a**	**MultiRC EM**	**ReCoRD F1**	**ReCoRD Accuracy**	**RTE Accuracy**	**WiC Accuracy**	**WSC Accuracy**
Previous best	84.4[d]	52.5[d]	90.6[d]	90.0[d]	88.2[d]	69.9[d]	89.0[d]
T5-Small	69.3	26.3	56.3	55.4	73.3	66.9	70.5
T5-Base	79.7	43.1	75.0	74.2	81.5	68.3	80.8
T5-Large	83.3	50.7	86.8	85.9	87.8	69.3	86.3
T5-3B	86.8	58.3	91.2	90.4	90.7	72.1	90.4
T5-11B	**88.1**	**63.3**	**94.1**	**93.4**	**92.5**	**76.9**	**93.8**

Results on SQUAD and S-GLUE (Source: Raffel et al., 2019)

BENCHMARK RESULTS

Model	WMT EnDe BLEU	WMT EnFr BLEU	WMT EnRo BLEU	CNN/DM ROUGE-1	CNN/DM ROUGE-2	CNN/DM ROUGE-L
Previous best	**33.8**[e]	**43.8**[e]	**38.5**[f]	43.47[g]	20.30[g]	40.63[g]
T5-Small	26.7	36.0	26.8	41.12	19.56	38.35
T5-Base	30.9	41.2	28.0	42.05	20.34	39.40
T5-Large	32.0	41.5	28.1	42.50	20.68	39.75
T5-3B	31.8	42.6	28.2	42.72	21.02	39.94
T5-11B	32.1	43.4	28.1	**43.52**	**21.55**	**40.69**

Results on MT/Summarization Benchmarks (Source: Raffel et al., 2019)

T5 - EXHAUSTIVE EXPERIMENTS

Conclusions

- Encoder-decoder architecture works best in this "text-to-text" setting
- More data, larger models & ensembling all boost the performance
  - Larger models trained for fewer steps better than smaller models on more data
  - Ensembling: Using same base pre-trained models worse than complete separate model ensembles
- Different denoising objectives work similarly well
- Updating *all* model parameters during fine-tuning works best (but expensive)