Project: Character-level LM[1]

Victoria Brami, Maxime Poli

[1] Jonathan H Clark et al., "Canine: Pre-training an efficient tokenization-free encoder for language representation".

What is CANINE?

Character-level language model.

- Sequence of Unicode code points $\rightarrow$ downsampling $\rightarrow$ deep transformer stack $\rightarrow$ upsampling + shallow transformer
- Two ways of pretraining: subword loss (CANINE-S) or autoregressive character loss (CANINE-C).
- Was evaluated on TyDiQA[2] and compared to mBERT[3]

[2] Jonathan H. Clark et al., "TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages".

[3] Devlin et al., "Bert: Pre-training of deep bidirectional transformers for language understanding".

What we have done

- Evaluation on other downstream tasks
  - Standard GLUE benchmark
  - Question answering on SQuAD v1.1
  - Multilingual on XNLI
- Assert robustness to noise

Standard baseline: GLUE

- Kept 6 among the 9 tasks (SST-2, MRPC, STS-B, QNLI, RTE, WNLI).
- Removed tasks with too large datasets.
- For the small datasets: high variance. WNLI problematic: some adversarial data.

Batch size: 6, learning rate: $1e-5$, 3 or 5 epochs depending on the task, sequence length: 2048 for CANINE, 512 for mBERT.

Standard baseline: GLUE

Results

Model	SST-2 (acc)	MRPC (F1/acc)	STS-B (pears./spear.)
mBERT reference	0.9232	0.8885 / 0.8407	0.8864 / 0.8848
mBERT	**0.8842**	**0.8920 / 0.8505**	**0.8838 / 0.8815**
CANINE-S	0.8165	0.8739 / 0.8260	0.8373 / 0.8364
CANINE-C	0.8394	0.8785 / 0.8284	0.8333 / 0.8336

Model	QNLI (acc)	RTE (acc)	WNLI (acc)
mBERT reference	0.9066	0.6570	0.5634
mBERT	**0.9107**	**0.6679**	0.5634
CANINE-S	0.8735	0.6173	0.5634
CANINE-C	0.8667	0.6282	0.5634

Table: Results on the dev set of 6 tasks from the GLUE benchmark. The results of "mBERT reference" are those given by the Transformers library.

Question answering: SQuAD

- Reading comprehension dataset
- In English ($\neq$ TydiQA which is multilingual)
- Restrict the fine-tuning on 60% of the dataset

Batch size: 6, learning rate: $1e-5$, 2 epochs, sequence length: 2048 for CANINE, 512 for mBERT.

Question answering: SQuAD

Results

Model	F1-Score	Accuracy
mBERT reference	0.8852	0.8122
mBERT	**0.8724**	**0.7966**
CANINE-S	0.7238	0.6165
CANINE-C	0.7023	0.5798

Table: Scores on the SQuAD Dataset. The reference model was trained on the full dataset, not on 60% of it as the other models.

Multilingual: XNLI

- Sentence pairs in 15 languages
- For a pair: determine whether the hypothesis entails, contradicts the premise, or none of them.
- Training in English; evaluation in Spanish, Deutsch and Vietnamese

Multilingual: XNLI

Results

Model	Spanish	Deutsch	Vietnamese
mBERT reference		0.7094	
mBERT	**0.7341**	**0.7028**	**0.6851**
CANINE-S	0.6526	0.5879	0.4878
CANINE-C	0.6578	0.6402	0.5124

Table: Accuracy obtained with mBERT and CANINE on XNLI on Spanish, Deutsch and Vietnamese languages. The reference results were given only for Deutsch.

Adding noise

Noise mechanism

Set proportion of sentences in which artificial noise is going to be applied. For each of those sentence, the number of words to perturb is chosen randomly.

Kind of Noise	**Example**
None	*His trial was moved to Virginia Beach.*
Subtract letter	*His trial was **oved** to Virginia Beach.*
Add letter	*His trial **wtas** moved to Virginia Beach.*
Swap letters	*His trial was moved **ot** Virginia Beach.*
Replace letters	*His trial was **moped** to Virginia Beach.*
Swap words	*His trial was moved **Virginia to** Beach.*

Table: Different sorts of noise applied on an input sentence from MRPC.

Adding noise

On GLUE

Used 10%, 20%, 40% and 90% of noise proportion. Full results in the report.

20% Noise			
Model	**SST-2 (acc)**	**MRPC (F1/acc)**	**STS-B (pears./spear.)**
mBERT	**0.8876**	**0.8958 / 0.8603**	**0.8639 / 0.8615**
CANINE-S	0.8177	0.8427 / 0.7868	0.8209 / 0.8178
CANINE-C	0.8326	0.8685 / 0.8211	0.8297 / 0.8300

Model	QNLI (acc)	RTE (acc)	WNLI (acc)
mBERT	**0.9021**	**0.7076**	0.4507
CANINE-S	0.8775	0.5957	0.4085
CANINE-C	0.8669	0.6420	0.4366

Table: Scores obtained on GLUE tasks with 20% of noise.

Adding noise

On GLUE

90% Noise			
Model	**SST-2 (acc)**	**MRPC (F1/acc)**	**STS-B (pears./spear.)**
mBERT	**0.8612**	**0.8702 / 0.8186**	0.8070 / 0.8048
CANINE-S	0.8039	0.8486 / 0.7892	0.7807 / 0.7854
CANINE-C	0.8326	0.8483 / 0.7966	**0.8108 / 0.8138**

Model	**QNLI (acc)**	**RTE (acc)**	**WNLI (acc)**
mBERT	**0.8766**	**0.6209**	0.4789
CANINE-S	0.8511	0.5776	0.4930
CANINE-C	0.8543	0.5884	0.5211

Table: Scores obtained on GLUE tasks with 90% of noise.

Adding noise

On GLUE

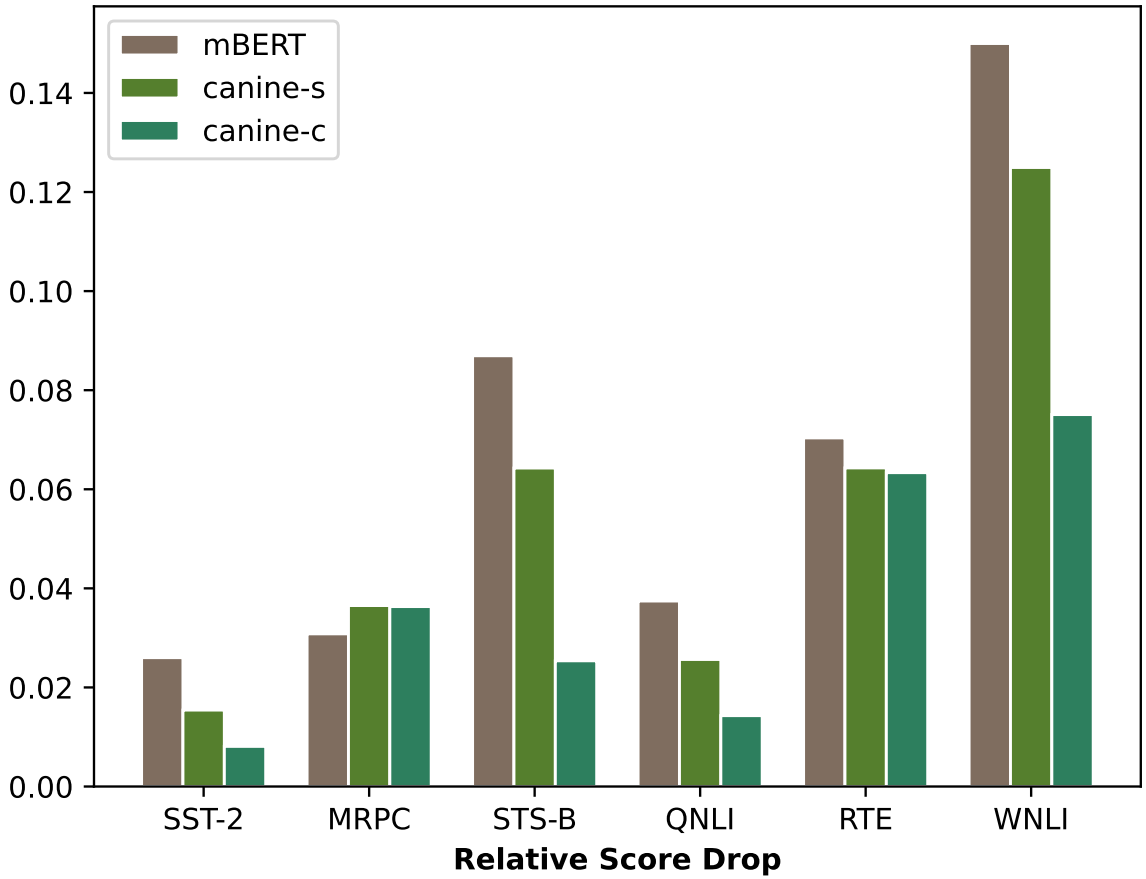

Figure: Relative evolution of the scores on GLUE tasks for CANINE and mBERT with 90% noised dataset. The relative drop is the highest for mBERT on almost all the tasks.

Adding noise

On GLUE

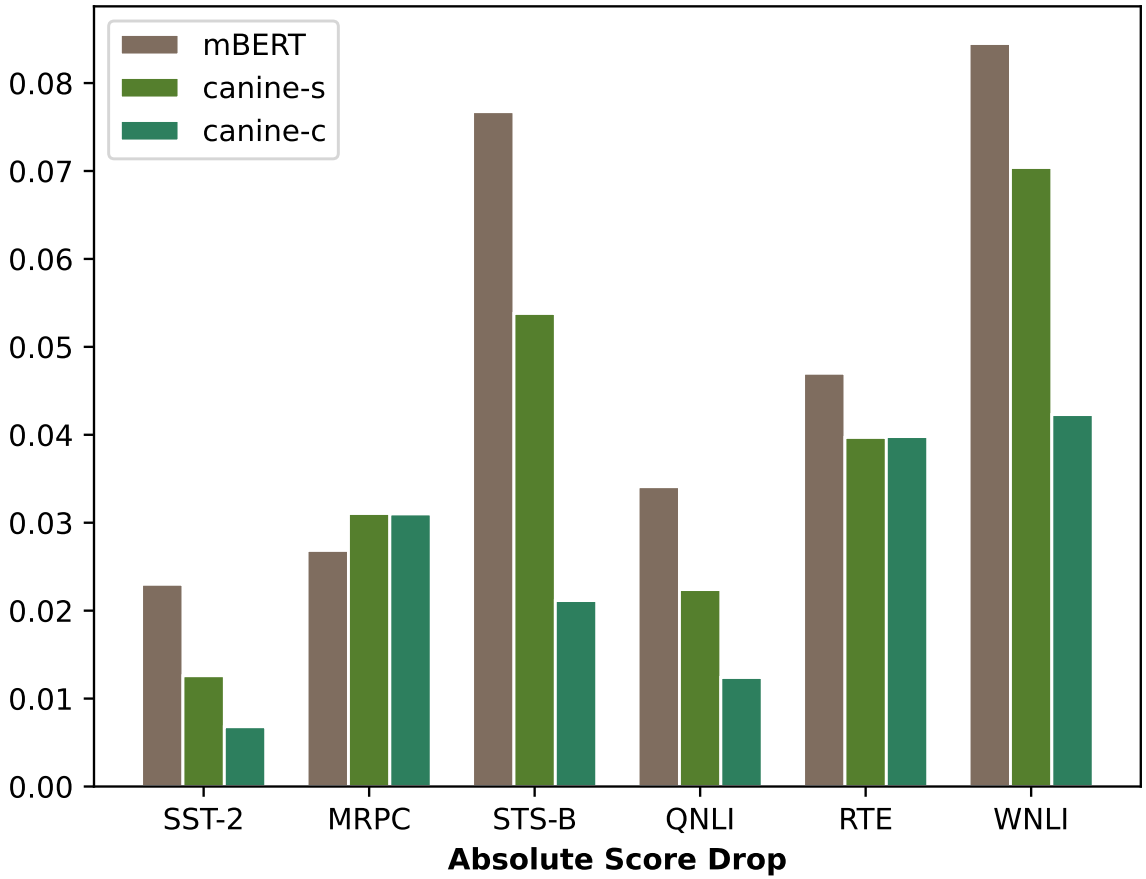

Figure: Absolute evolution of the scores on GLUE tasks for CANINE and mBERT with 90% noised dataset.

Adding noise

On XNLI

Model	Spanish	Deutsch	Vietnamese
mBERT	**0.6799**	**0.6494**	**0.6361**
CANINE-S	0.6205	0.5655	0.4691
CANINE-C	0.6390	0.6064	0.4920

Table: Accuracy obtained on XNLI task when taking 70% of noise proportion. The scores drop significantly.

Adding noise

On XNLI

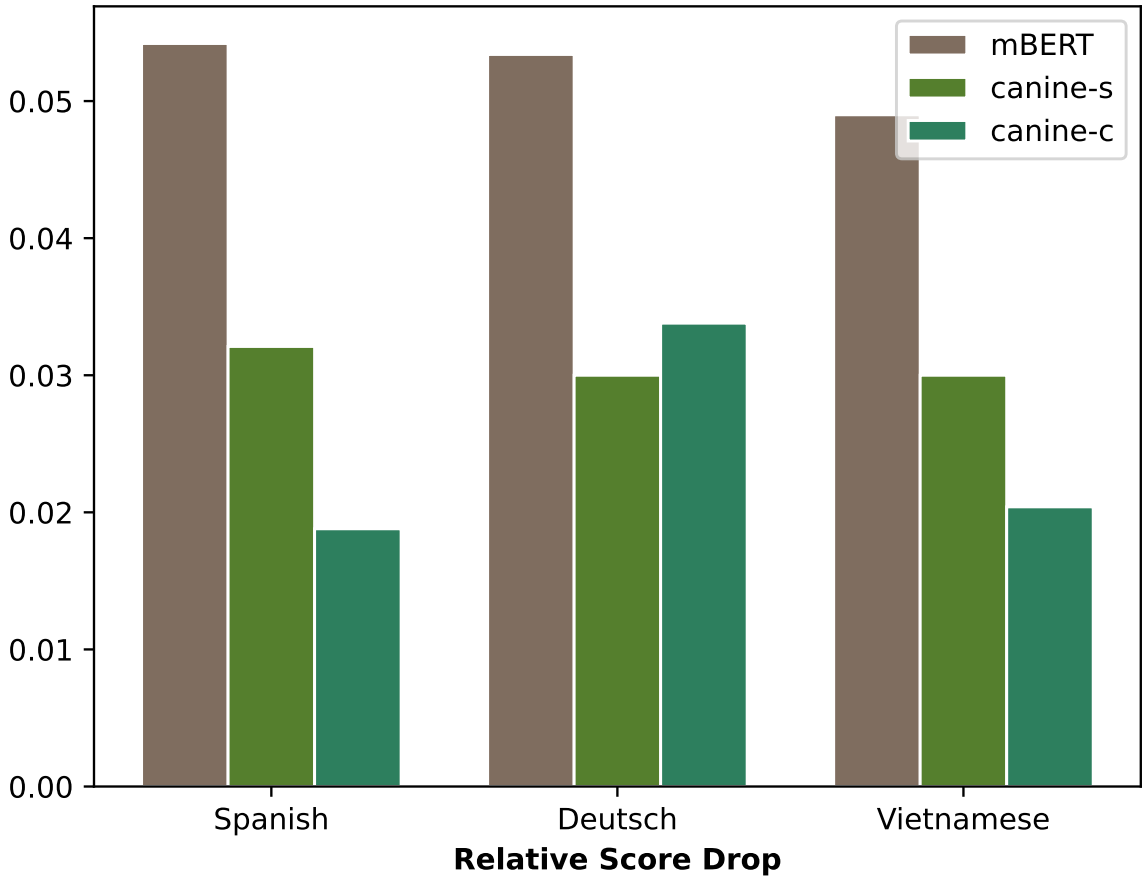

Figure: Relative evolution of the scores on XNLI task for CANINE and mBERT on respectively Spanish, Deutsch and Vietnamese languages (with 70% noised dataset).

Adding noise

On XNLI

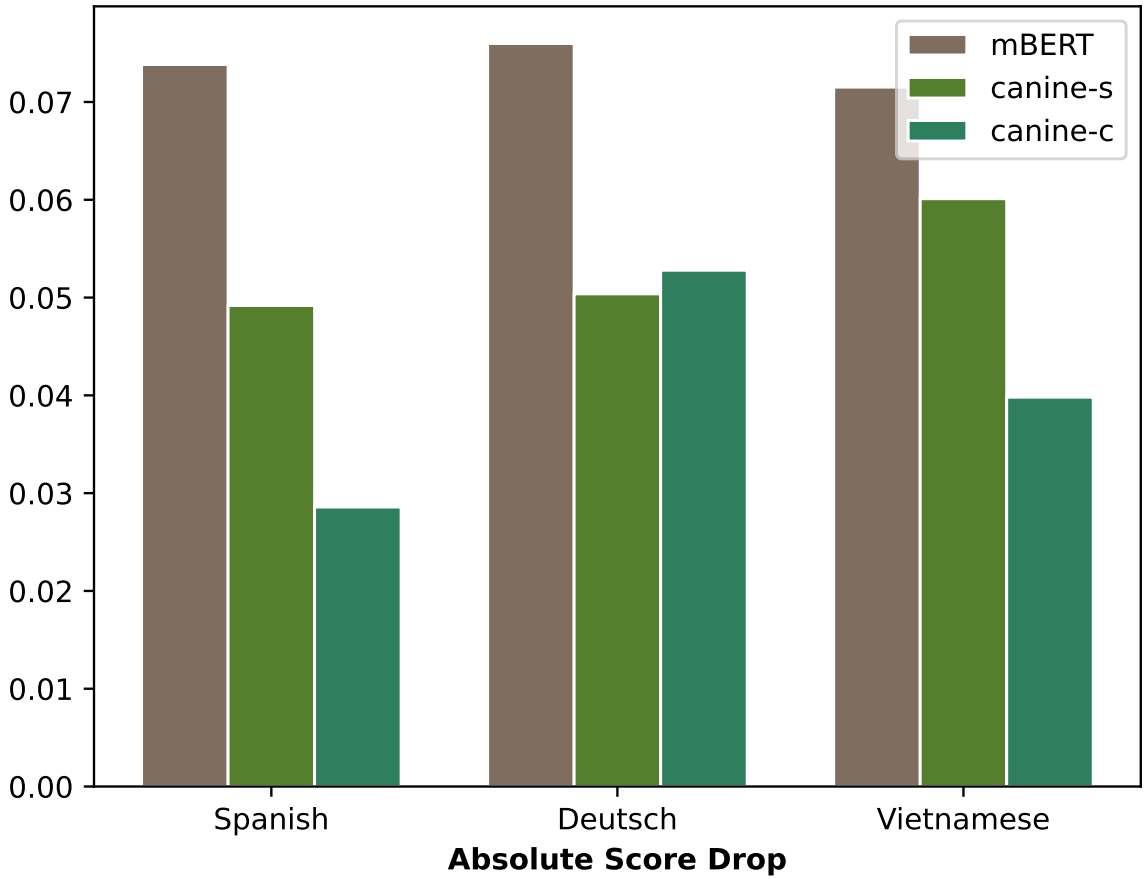

Figure: Absolute evolution of the scores on XNLI task for CANINE and mBERT on respectively Spanish, Deutsch and Vietnamese languages (with 70% noised dataset).

Conclusion

What we found:

- For other benchmarks than TydiQA: CANINE is outperformed by mBERT.
- But CANINE seems to be a bit more robust to the addition of noise, even if mBERT is always better.

What can be done:

- Evaluate with real world noise: text from social media. But cannot control the proportion of noise.
- More runs to reduce the impact of stochasticity on the results.
- Find a way to add noise on languages that do not use the Latin script.

References

[1] Jonathan H Clark et al. "Canine: Pre-training an efficient tokenization-free encoder for language representation". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 73–91.

[2] Jonathan H. Clark et al. "TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages". In: *Transactions of the Association for Computational Linguistics* (2020).

[3] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).