Project report: Project 4 - Character-level LM

Victoria Brami, Maxime Poli
victoria.brami@eleves.enpc.fr, maxime.poli@eleves.enpc.fr

1. Introduction

Tokenization is one of the core components of the NLP pipeline: its crucial purpose is to translate text into data that can be processed by a language model. It can be done, as with WordPiece [15], by splitting the text into words and the rarer words into smaller meaningful subwords: this is subword tokenization. This approach is used in BERT[4] and other Transformer[10] models based on it, such as DISTILBERT[7] or MOBILEBERT[9].

However, such tokenization may lead the models to be sensitive to the noise in the training data, would it be naturally present [8] or adversarially created [5]. Moreover, splitting sentences into subwords may work well in English, but it is not as adapted for other languages with a different morphology.

In order to tackle this issue, character-level language models such as CANINE [2] have been proposed. In [2] the authors introduce this new model which they have pre-trained and then evaluated on TYDI QA [1], and compared it to mBERT.

In this project, we wanted to do a more extensive analysis of the performances of CANINE on downstream tasks, and to evaluate its robustness to noisy inputs. We fine-tuned and evaluated both versions of CANINE on the GLUE[11] benchmark and compared them to mBERT. We then studied their performances on multilingual datasets. Finally, we handcrafted a simple algorithm to add noise to the input data, and we evaluated the robustness of CANINE using the previous datasets.

2. CANINE

CANINE is a character-level model which differs as least as possible from deep transformer models such as (m)BERT. What makes the greatest difference is that it does not use an explicit tokenization step. Instead, the model is trained directly at a Unicode character-level: the text is turned into a sequence of characters which are converted into its Unicode code point.

However, training at a character-level inevitably comes with a longer sequence length, which CANINE solves with an efficient downsampling strategy, before applying the encoder. This latter portion of the model is similar to mBERT and derivatives. Finally, a character-level output representation is built by upsampling and then applying a final transformer layer. CANINE can be pre-trained using either a subword loss (those models are CANINE-S) or an autoregressive character loss (denoted as CANINE-C).

3. Experiments

3.1. Experimental setup

We fine-tuned our models on two separate NVIDIA P100 GPUs using Google Colab. As it is important for CANINE to have a long sequence length to capture enough context, we could not use large batch sizes. In the end, we kept a maximum sequence length of 2048 for CANINE models and we used 512 for mBERT. The maximum batch size that could be used for CANINE was 6, and we kept the same size for mBERT in order to have similar experimental setups. Similarly as in the original paper, we observed that mBERT was 35% faster to train than CANINE.

We used the pre-trained models available in the Transformers[14] library. The available mBERT has been pre-trained on the multilingual Wikipedia data alone, while the CANINE models were pre-trained on Wikipedia+BookCorpus. This is different from [2], as they pre-trained mBERT also on Wikipedia+BookCorpus in order to have fair comparisons. One could therefore expect that our mBERT would perform worse than the one studied in the original CANINE paper.

We originally wanted to reproduce the published results on TYDI QA, but with limited computational resources, both in terms of GPU capabilities and maximum training duration, it was not feasible for us.

3.2. Standard baseline

GLUE We first evaluate CANINE models on standard tasks. To do so, we choose various sub-tasks of the General Language Understanding Evaluation (GLUE) benchmark [12], which are for most Sequence Classification tasks. We found relevant to use these tasks as GLUE benchmark is very commonly used to measure language models efficiency. Among the nine tasks available in GLUE, we do not

evaluate the models on QQP and MNLI which are too large, and neither on CoLA since the task of English acceptability is not compatible with the study of the influence of the noise that we do later in this work.

We follow the training procedure advised in the Transformers library[1]. We fine-tune CANINE-S, CANINE-C and mBERT for 3 epochs on each task except for MRPC and WNLI, where the fine-tuning is performed over 5 epochs because the two datasets are much smaller. We also use a learning rate of $1e-5$ for the 3 models and the hyperparameters discussed earlier. The performances of the three models are displayed in table 1.

Some of these tasks have a small dataset and training can lead to high variance in the results between different runs, but we find similar ones as those of reference in the Transformers library obtained with larger batches. Therefore we can make the assumption that our training procedure is correct and that our small batch size does not significantly influence the final results. Furthermore, the WNLI set is problematic since the split between the train and dev set is adversarial, and results on this task should not influence the overall appreciation of the models[2].

We find that both CANINE-C and CANINE-S consistently slightly underperform mBERT across the different tasks.

3.3. Question Answering baseline

Because we could not replicate the results on TYDI QA, a question answering dataset covering 11 diverse languages, we instead focused on evaluating CANINE on a different question answering dataset and on a multilingual task.

SQuAD v1.1 The Stanford Question Answering dataset (SQuAD)[6] is a reading comprehension dataset in English, commonly used to evaluate deep encoders performances. Given a full paragraph of an article, it must return the start and end bytes of the answer to a specific question. We restrict the fine-tuning on 60% of SQUAD dataset, and we train for 2 epochs with a learning rate of $1e-5$, and the same other hyperparameters as before.

Using this training procedure yields the results in table 2. Again, our results with mBERT are close to the ones given by the Transformers library. This time the drop in performances between mBERT and both CANINE models is significant.

3.4. Multilingual benchmark

We then evaluate the character-level tokenization approach used for CANINE on a multilingual task, to seek if such a model would generalize better on different languages than those with a classical subword-tokenizer.

XNLI The Cross-lingual NLI Corpus (XNLI)[3] is a subset of MultiNLI[13], extended to 15 languages and made of sentence pairs annotated with textual entailment. The task is a 3-way classification task: given a pair of sentences (premise and hypothesis), the goal is to determine whether the hypothesis entails, contradicts the premise, or none of them. The training is done with English, and we restrict the evaluation to one of three languages: two languages using the Latin script (Spanish and Deutsch) and Vietnamese.

We deliberately choose languages which alphabet is not made of logograms. As we later analyze the effect of incorporating noise into the dataset, the noise cannot be modelled in the same way for logograms-based languages as previously.

Table 3 displays the scores obtained for these three languages. One can notice that the more the validation language used is morphologically close to English, the higher the accuracy is: the accuracies reach their highest value for all models on Spanish validation dataset. On the contrary, they struggle more on the same task when the validation dataset is quite semantically different from English language. We also notice a bigger gap between mBERT and CANINE on a language like Vietnamese.

4. Adding noise

4.1. Noise mechanism

One of the motivations that lead to the design of CANINE was to find an approach that can "generalize beyond the orthographic forms encountered during pre-training"[2]. Thanks to the character level tokenization, CANINE may be more robust to orthographic alterations. In order to assess this hypothesis, we devised a simple mechanism to add noise to existing data. It consists in either subtracting, replacing, swapping or adding a letter in a word, or to swap two words. Some examples of such altered sentences are shown in table 4.

With this simple scheme we can control the level of noise that we want. We fine-tune again CANINE and mBERT on the previous datasets but this time with varying levels of noise. A percentage of noise correspond to the proportion of sentences where noise is applied. Each noised sentence of N words has at most N perturbations, the number of perturbations within the sentence being chosen randomly. We use 10%, 20%, 40% and 90% of noise on GLUE, and 70% on XNLI. For a given task or target language, the noise process has been applied just once: mBERT and both CANINE models have seen the same training and evaluation data. For both datasets, we use the same training hyperparameters as before.

[1] https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/README.md

[2] See entry (12) in https://gluebenchmark.com/faq.

Model	SST-2 (acc)	MRPC (F1/acc)	STS-B (pears./spear.)	QNLI (acc)	RTE (acc)	WNLI (acc)
mBERT reference	0.9232	0.8885 / 0.8407	0.8864 / 0.8848	0.9066	0.6570	0.5634
mBERT	**0.8842**	**0.8920 / 0.8505**	**0.8838 / 0.8815**	**0.9107**	**0.6679**	0.5634
CANINE-S	0.8165	0.8739 / 0.8260	0.8373 / 0.8364	0.8735	0.6173	0.5634
CANINE-C	0.8394	0.8785 / 0.8284	0.8333 / 0.8336	0.8667	0.6282	0.5634

Table 1: Results on the dev set of 6 tasks from the GLUE benchmark. The results of "mBERT reference" are those given by the Transformers library.

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

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

In addition to the raw scores, we also tracked the relative and absolute drop in performances. For a given noise percentage, for each model independently, we computed the relative and absolute evolution of the scores compared to those obtained without any noise.

4.2. Results

GLUE with noise The results on noisy GLUE are shown in table 5. Overall, there may be artefacts due to the stochasticity of our noise mechanism, and the results would have benefited from making several runs and taking the mean and standard deviation of the scores, especially on the smaller tasks.

We plot in figs. 1 and 2 the relative and absolute drops in scores with 90% noised data. In all GLUE tasks, fig. 1 illustrates that CANINE-C demonstrates a higher robustness than mBERT towards noise, but as it can be seen in table 5 mBERT still has better scores overall. This gap in scores is quite visible on SST-2, STS-B and QNLI tasks. Even though the absolute scores are higher for mBERT, CANINE models may tend to generalize better on noisy datasets. It is still important to highlight that the differences in robustness between the models in only in the order of a few percents.

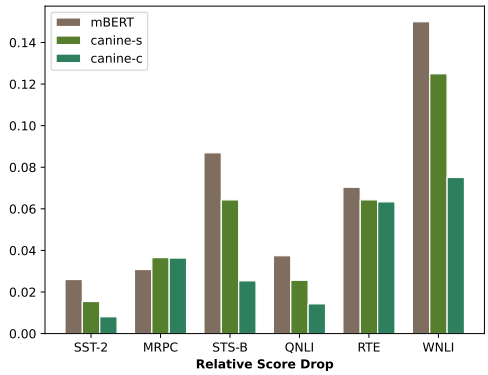

Figure 1: Relative evolution of the scores on GLUE tasks for CANINE and mBERT with 90% noised dataset. The relative drop is the highest for mBERT on almost all the tasks.

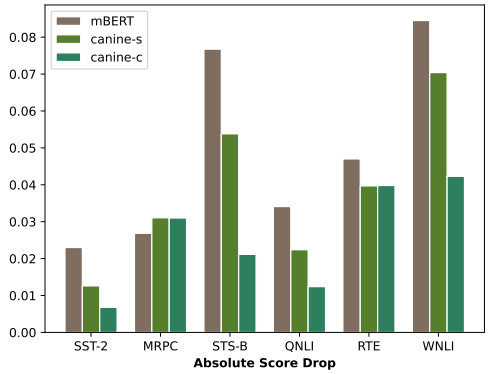

Figure 2: Absolute evolution of the scores on GLUE tasks for CANINE and mBERT with 90% noised dataset.

XNLI with noise For XNLI, we choose a 70% noise proportion, in order to measure the effect of adding a high level of noise to multilingual data. As shown in table 6, the fine-tuning on all the 3 languages is impacted by the addition of noise in the training dataset. We observe a decrease of 4-5 % for all scores on all models. Again, as seen in figs. 3 and 4 the relative and absolute decrease in accuracy is larger

Kind of Noise	Example
None	*Forcing his way into country properties was possible due to his legitimation.*
Subtract letter	*Forcing his way into country **proerties** was possible due to his legitimation.*
Add letter	*Forcing his way into **countery** properties was possible due to his legitimation.*
Swap letters	*Forcing his way into country properties was possible **deu** to his legitimation.*
Replace letters	*Forcing his way into country properties was **possible** due to his legitimation.*
Swap words	*Forcing his way into **properties country** was possible due to his legitimation.*

Table 4: Different sorts of noise applied on GLUE tasks datasets' words.

10% Noise						
Model	**SST-2 (acc)**	**MRPC (F1/acc)**	**STS-B (pears./spear.)**	**QNLI (acc)**	**RTE (acc)**	**WNLI (acc)**
mBERT	**0.9048**	**0.9101 / 0.8750**	**0.8714 / 0.8681**	**0.9043**	**0.6931**	0.5634
CANINE-S	0.8268	0.8470 / 0.7892	0.8300 / 0.8307	0.8625	0.5957	0.5634
CANINE-C	0.8612	0.8305 / 0.8039	0.8333 / 0.8336	0.8720	0.6245	0.5634
20% Noise						
Model	**SST-2 (acc)**	**MRPC (F1/acc)**	**STS-B (pears./spear.)**	**QNLI (acc)**	**RTE (acc)**	**WNLI (acc)**
mBERT	**0.8876**	**0.8958 / 0.8603**	**0.8639 / 0.8615**	**0.9021**	**0.7076**	0.4507
CANINE-S	0.8177	0.8427 / 0.7868	0.8209 / 0.8178	0.8775	0.5957	0.4085
CANINE-C	0.8326	0.8685 / 0.8211	0.8297 / 0.8300	0.8669	0.6420	0.4366
40% Noise						
Model	**SST-2 (acc)**	**MRPC (F1/acc)**	**STS-B (pears./spear.)**	**QNLI (acc)**	**RTE (acc)**	**WNLI (acc)**
mBERT	**0.8807**	**0.8817 / 0.8382**	**0.8450 / 0.8423**	**0.8968**	**0.6390**	0.5070
CANINE-S	0.8452	**0.8817 / 0.8382**	0.7933 / 0.7958	0.8526	0.5776	0.4930
CANINE-C	0.8469	0.8541 / 0.8015	0.8115 / 0.8162	0.8523	0.5884	0.5070
90% Noise						
Model	**SST-2 (acc)**	**MRPC (F1/acc)**	**STS-B (pears./spear.)**	**QNLI (acc)**	**RTE (acc)**	**WNLI (acc)**
mBERT	**0.8612**	**0.8702 / 0.8186**	0.8070 / 0.8048	**0.8766**	**0.6209**	0.4789
CANINE-S	0.8039	0.8486 / 0.7892	0.7807 / 0.7854	0.8511	0.5776	0.4930
CANINE-C	0.8326	0.8483 / 0.7966	**0.8108 / 0.8138**	0.8543	0.5884	0.5211

Table 5: Scores obtained on GLUE tasks with different proportion of noise.

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

for mBERT.

5. Conclusion

In this work we evaluated CANINE of standard benchmarks that were not considered in the original article, and we found that, for a given training procedure, it was systematically slightly surpassed by mBERT. We finally evaluated the robustness of CANINE and mBERT to the addition of artificial noise, and found that even if mBERT kept better performances overall, the drop in scores for both CANINE models is lower. This might indicate that CANINE is more robust to this kind of noise.

In this work, we did not reproduce the results of [2] on TYDI QA, which could have strengthen the confidence in our experimental setup and in our subsequent observations, since this was the only downstream task that was originally considered. Similarly, doing several runs when estimating the robustness to noise would have reduce the impact of the stochasticity of the mechanism on smaller datasets.

Furthermore, we did not consider languages with a writing system different from the Latin script[3], and our noise mechanism would not be adapted to theme. For instance, we cannot model noise in Japanese by swapping two characters or pictograms. Another path to investigate would be to assess the robustness to noise in real world conditions. One would not be able to control the desired level of noise anymore, but comparing CANINE and mBERT on text from

[3]The Vietnamese dataset in XNLI uses the Vietnamese alphabet which is the Latin writing script for Vietnamese.

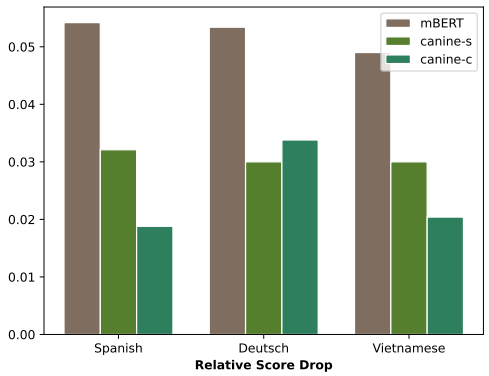

Figure 3: Relative evolution of the scores on XNLI task for CANINE and mBERT on respectively Spanish, Deutsch and Vietnamese languages (with 70% noised dataset). The relative accuracy drop is lower for CANINE models.

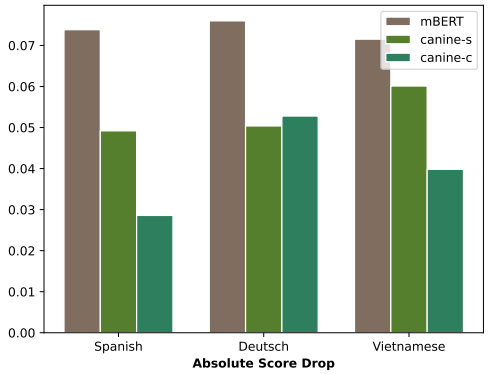

Figure 4: Absolute evolution of the scores on XNLI task for CANINE and mBERT on respectively Spanish, Deutsch and Vietnamese languages (with 70% noised dataset).

social media may be another way to test prediction quality on noisy data.

References

[1] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 2020. 1

[2] Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91, 2022. 1, 2, 4

[3] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. 2

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[5] Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. Combating adversarial misspellings with robust word recognition. *arXiv preprint arXiv:1905.11268*, 2019. 1

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016. 2

[7] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. 1

[8] Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985*, 2020. 1

[9] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic BERT for resource-limited devices. *CoRR*, abs/2004.02984, 2020. 1

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[11] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 1

[12] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2018. 1

[13] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017. 2

[14] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. 1

[15] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun,

Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016. 1