CANINE: Pre-training an Efficient Tokenization-Free Encoder for Language Representation

Gabriel Watkinson
ENSAE
gabriel.watkinson@ensae.fr

Marine Astruc
ENS Paris-Saclay
marine.astruc@ens-paris-saclay.fr

Josselin Dubois
ENS Paris-Saclay
josselin.dubois@ens-paris-saclay.fr

Javier Ramos-Gutiérrez
ENS Paris-Saclay
javier.ramos_gutierrez@ens-paris-saclay.fr

1 Introduction

The CANINE model (Clark et al., 2021) is an innovative language model that has been trained using a tokenization-free encoding method, leveraging the Unicode Standard to process raw strings. This approach enables the generation of accurate and comprehensive language representations without the need for explicit tokenization, making it particularly suitable for languages with complex morphology or non-standard orthography where expert knowledge is often required to produce the best models. In the field of natural language processing (NLP), CANINE represents a significant advancement, offering a more efficient and precise approach to pre-training language representations.

In this report, we present an experiment conducted to explore and validate the capabilities of the CANINE model. Our objective was to investigate its performance across various NLP tasks, comparing it to the well-known multilingual BERT model that serves as the baseline for CANINE. Additionally, we examined its multilingual capabilities, particularly in languages with specific tokenization requirements.

The experiment was divided into four parts, each focusing on a specific task.

- First, we reproduced experiments on the Information-Seeking QA task. [3]
- Next, we replicated experiments on Named Entity Recognition. [11; 12; 1]
- Then, we delved into classification with Entailment Analysis on XNLI. [5]
- Finally, we explored Seq2Seq tasks with translation on the WMT14 dataset. [2]

Please note that all our code and experiences are available at https://github.com/gwatkinson/mva_snlp_canine for reference.

2 The CANINE model

Since BERT (Devlin et al., 2019), transformer based architecture have become state of the art in the NLP field. CANINE tries to replicate such an architecture, but in a tokenizer-free fashion, meaning that it works directly at the character level. The inputs are therefore the sequence of Unicode characters. However, a limitation of transformers are the quadratic cost in the length of the sequence and processing 512 tokens sequence is already quite expensive. Natural sentences have thousands of characters, it is unfeasible to use a transformer directly.

CANINE combines downsampling, which reduces the input sequence length using strided convolutions, allowing to capture local patterns and dependencies at the character level, with a deep transformer stack, which encodes context, resulting in speeds comparable to vanilla BERT and performs no tokenization on the input, avoiding the lossy information bottleneck and complex task, while keeping most of the advantages of transformers. It also allows to work on uncommon writing system (lowish resources languages) and can even work with emojis for example. See Figure 2 for a schema of the architecture.

The authors claim that CANINE is compared to other common language models such as BERT on several downstream tasks, such as question answering, natural language inference, sentiment analysis, and named entity recognition. And achieves competitive results on other tasks and languages, especially those that benefit from character-level information or have limited subword coverage.

3 Information Seeking

3.1 Dataset

For the Information Seeking task, we used the primary task subset of the TyDi QA dataset (Clark

	CANINE-C		CANINE-S	
	Ours	Paper	Ours	Paper
Exact matches	19%	N/A	21%	N/A
F1-Score	**72.9**	65.7	**76.1**	66.0

Table 1: Exact matches and F1-scores for the QA task, for both our training and the results presented in the paper.

	mBERT		CANINE-C	
	Ours	Paper	Ours	Paper
CoNLL	**89.3**	87.8	**88.5**	74.0
MasakhaNER	60.2	**72.4**	52.2	**65.5**

Table 2: Average F1 scores on the NER tasks, evaluated on the CoNLL and the MasakhaNER datasets.

et al., 2020). This dataset contains information-seeking questions in 11 typologically diverse languages (Japanese, Arabic, English, ...), which is great for testing the adaptability of the tokenizer-free aspect of CANINE. In this dataset, there is a set of questions associated to a context containing the answer and indexes of the start and the end of a probable answer.

3.2 Model and Methodology

To do this task, we only use a part of the dataset because the entire dataset has too much data to process with our computers. We nevertheless made sure to have all the languages represented. We leverage the utilities from HuggingFace (Wolf et al., 2020), especially the CanineForQuestionAnswering module.

The model was only trained for 4 epochs, first because we can't train it longer with our GPU credits, but also because it was enough since the model was pre-trained. We trained both pretrained models CANINE C and CANINE S to compare them to the results presented in the paper.

3.3 Results and Limitations

The evaluation is done with the SQuAD metric, which is the scoring script for the first version of the Stanford Question Answering Dataset (SQuAD) introduced by Rajpurkar et al. (2016). This metric measures the average overlap between the prediction and ground truth answer.

Results are summarized on 1. Our results are surprisingly higher than the f1-scores presented on the paper. This is a strange phenomenon that is most likely explained by the fact that we only used a small part of the dataset. Moreover, CANINE S is more efficient than CANINE C in our case, whereas it is the opposite in the paper. This may be due to the fact that we only trained for few epochs, the behaviour may be reversed after convergence of the training.

4 Named Entity Recognition

4.1 Dataset

For the Named Entity Recognition task, two different kinds of datasets have been used. First, the CoNLL NER datasets in Spanish and Dutch (Tjong Kim Sang, 2002) and English (Tjong Kim Sang and De Meulder, 2003), which include labeled text from the newswire domain. Then, and in contrast with these European languages, we also use MasakhaNER (Adelani et al., 2021), which include annotated local news text in 10 African languages.

4.2 Model and Methodology

Similarly to the previous one, this task is fully supported by the HuggingFace utilities. In this case, CanineForTokenClassification is the module that we mostly used. Same as in the paper, we use CANINE-C and multilingual BERT as a baseline. The models were trained for 2 epochs. For the full specification of the hyperparameters, check the code in the GitHub repository.

4.3 Results and Limitations

For this task, the evaluation has been done using the regular F1 metric across all the classes. A summary of the results are shown in Tables 2 and 4. These follow a strange dual behavior.

For the European languages, every language exceeds the expected results by 10 points. Even if the German dataset was not included in the test, these results are unexpected. This may happen due to a slight mismatch between the beginning and intermediate classes on our side.

On the other hand, for the African languages, the general trend is to register a drop of 5-10 points in both mBERT and CANINE-C. However, we must remark sharp declines in the Amharic and Luo datasets, as well as an increase in the Nigerian Pidgin one. The latter can be somehow explained given its similarity to English, thus performing like the previous European languages.

Model	Train languages	English	Bulgarian	German	Greek
BERT	**0.673**	**0.707**	0.653	**0.617**	**0.597**
Canine-C	0.667	0.703	**0.667**	0.475	0.474
Canine-S	0.654	0.676	0.658	0.458	0.447

Table 3: F1 score on different test sets. The F1 score is a weighted average between the 3 class and languages for the first column, and is evaluated on a test set of 5k observations never seen during training.

5 Entailment Analysis

In this section, we will present the experiments we conducted to explore the classification capabilities of CANINE, something that is not done in the original paper.

5.1 Dataset

We used the XNLI dataset (Conneau et al., 2018), which is a subset of a few thousand examples from Multi-Genre Natural Language Inference (MultiNLI), meticulously translated into 14 different languages, including some with limited linguistic resources. The primary objective of the task is to determine textual entailment, involving the classification of two given sentences into one of three categories: implication, contradiction, or neutrality.

5.2 Experimental setup

Since NLI is just a classification task, we added a simple classification head on top of the language model. In our case, we only looked at CANINE, for which we used the pretrained weights obtained with the procedure describe in the paper, CANINE-C using the character loss and CANINE-S using the subword loss. We used mBERT for our baseline, since both architectures are made to be similar.

We had to reduce the size of the dataset because of limited resources, we thus decided to select less than 300k observations across a few languages. The details of the runs can be found on our GitHub.

5.3 Results

First experiment We trained the three models on a subset of 300k observations containing English, French, Spanish, Bulgarian and Russian. The table 3 shows the results, and the Figure 3 contains the details. On average across the languages used during training, the performance is really similar, with CANINE-S falling behind. The only language where CANINE outperforms BERT is Bulgarian. On the other hand, for German and Greek, that were not in the train set, we expected that CANINE would transfer better, but it is not the case and BERT greatly outperforms it. This might be linked to the pretraining, indeed, if they are differences in the procedure to obtain the pretrained weights, it's difficult to conclude on the objective performance of the models. To do that, we would need to retrain from scratch.

Second experiment We observe the same phenomenon in another experiment using Arabic and Turkish in the training set. BERT still transfers better to languages like Urdu, Swahili, Hindi or Spanish (see Figure 4).

Third experiment Lastly, Clark et al. (2021) mentioned that they expected their model to be more robust to character level perturbation. We used the *nlpaug* library (Ma, 2019) to augment the test dataset with random character deletions and swaps, as well as extra random spaces. We expected that the CANINE models would have the best performance as it can somewhat make abstraction of missing characters or extra spaces thanks to its architecture. However, the Figure 5 shows that this is not the case and BERT still is the better model. This figure also shows the performance gains when using more data.

Finally, the Table 6 presents some examples of the French model using 300k observations. It also displays some perturbed inputs and the change in predictions.

Conclusion Our results suggest that BERT is slightly more effective for NLI tasks, especially when transferring from one language to another. This might be because of the difference in the architectures, but we highly suspect that they are also differences in the pretraining and on the thoroughness of the fine-tuning, that is likely better for BERT, but we can't conclude on this with only our experiments. When comparing the C and S pretraining strategies, we see that CANINE-C seems to outperform CANINE-S in most situations, but the difference is not huge.

6 Translation

6.1 Dataset

For the translation task, we used the WMT14 dataset (Bojar et al., 2014), based on statmt.org.

This dataset is particularly suitable for a tokenizer-free model as CANINE as it has various languages (French, Czech, Hindi) with very different alphabets and structures.

6.2 Model and Methodology

As CANINE is not designed for Seq2Seq tasks, we had to adapt its use. We used CANINE as an encoder (to use its features) on which we stack a decoder for translation.

To do this, we used the EncoderDecoder model of HuggingFace to use a pre-trained decoder. In particular, we used BART (Lewis et al., 2019). As the model was too big, we could not train it on kaggle or google collab. So we had to freeze the layers of either the encoder or the decoder. We chose to freeze CANINE, for several reasons. First, the decoder is trained with its own tokenizer. Since the goal of CANINE is to do without a tokenizer, we need to fine-tune the decoder to make it forget its embedding. Moreover, CANINE's paper advocates its pre-trained features. So it makes sense to use it in the fine-tune.

Since CANINE works on Unicode characters (i.e. 1,114,112 characters), the decoder embedding has a shape of (1.114M, hidden_size). Once again, this embedding is too big to be stored on our machines, and was crashing Kaggle and Google Colab. The only way to overcome this problem is to restrict the embedding to the first Unicode characters (mainly Latin characters). Although the model works this way, we lose the interest of CANINE which is to be able to adapt to any language.

6.3 Results and Limitations

Even though we couldn't run it, the notebook of this task is available on our GitHub. We planned to train on translating from French, Czech or Hindi to English as it shows a large diversity of characters.

However, results can be obtained by evaluating the model only on European languages (which reduce the vocabulary size and then the dimension of the embedding, since we can limit ourselves to the first Unicode characters). As the model is big and the dataset large, the training was long, and we were not able to train it for more than two epochs on GPU. Note that it is pretty reasonable, as we're only doing fine-tuning.

By looking at inferences on table 5, we can see that despite the lack of training, results are promising even though not really good. The model is able to capture some meanings and to generate real

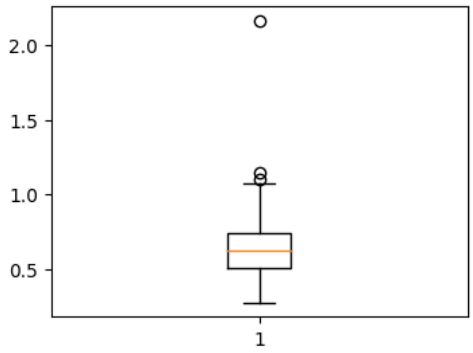

Figure 1: Bleu score on 216 samples

words, even thought it works on characters (it can capture the structure of a dialogue for example, and most of the words are correct in French).

Nevertheless, we can see few words that are not French. For example, "compagniers" or "souvert" in the second inference in table 5. This is a problem that is less likely to append with vocabulary-based models, even though we can expect CANINE to prevent those errors with a better training. We'll not really comment on the translation performances because of the lack of training, but none of the sentences were correctly translated. We obtain a mean BLEU score of less than 1% (see figure 1). Moreover, since the model has to generate its output character after character, the inference time is way longer than the one of a vocabulary-based model.

7 Discussion and Conclusion

Even though there exists some tokenizers that aim to work for every language, such as SentencePiece (Kudo and Richardson, 2018) used in ALBERT for example. Being tokenizer-free is a very important step for having models that can adapt to every language, as different as they may be, especially to build end-to-end models.

CANINE shows very good results on a wide variety of tasks, as we have seen in Sections 3, 4 and 5, and a great capability of adaptation to different languages. However, there's still limitations of being tokenizer-free with current models for some tasks, in sequence generation for instance (Sec. 6), as it has a high memory footprint.

8 Contributions

• The ideas for the tasks came after a discussion during the first week of the project.

• Marine Astruc did the experiments for the QA task.

• Javier Ramos-Gutiérrez worked on the NER task.

• Gabriel Watkinson did the experiments on the NLI task and helped structure the LaTeX report and the GitHub.

• Josselin Dubois worked on the translation task, helped for the QA task as it ended up being harder than expected. He also played an important part in the presentation.

• Lastly, the conclusion and discussion was a common effort between all four of us regrouping elements from all experiments.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. 2021. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131. 1, 2

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics. https://huggingface.co/datasets/wmt14. 1, 3

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

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. CANINE: pre-training an efficient tokenization-free encoder for language representation. *CoRR*, abs/2103.06874. 1, 3, 6

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://huggingface.co/datasets/xnli. 1, 3

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 1

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. 4

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. 4

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug. 3

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. https://huggingface.co/spaces/evaluate-metric/squad. 2

Erik F. Tjong Kim Sang. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan. 1, 2

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada. 1, 2

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. 2020. 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. Association for Computational Linguistics. 2

Appendices

A Additional figures and tables

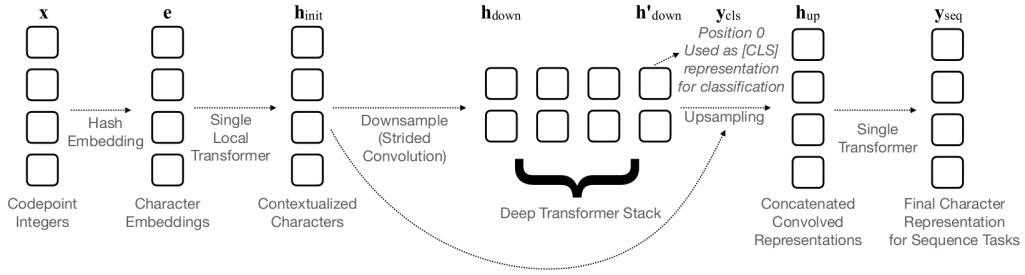

Figure 2: Schema of CANINE's neural architecture. Illustration taken from the original paper (Clark et al., 2021)

B Additional material for the NER experiment

	mBERT		CANINE-C	
	Ours	Paper	Ours	Paper
CoNLL				
Dutch	90.3	90.2	87.0	74.7
English	90.3	91.1	89.9	79.8
German	-	82.5	-	64.1
Spanish	87.1	87.6	88.6	77.4
Macro Avg	89.3*	87.8	88.5*	74.0
MASAKHANER				
Amharic	0.0	0.0	15.6	44.6
Hausa	78.2	89.3	69.7	76.1
Igbo	76.5	84.6	69.6	75.6
Kinyarwanda	61.7	73.9	45.3	58.3
Luganda	64.7	80.2	59.9	69.4
Luo	27.6	75.8	15.7	63.4
Nigerian Pidgin	82.7	89.8	71.1	66.6
Swahili	83.0	87.1	68.2	72.7
Wolof	57.8	64.9	54.6	60.7
Yorùbá	69.3	78.7	52.4	67.9
Macro Avg	60.2	72.4	52.2	65.5

Table 4: Language-wise breakdown for NER task for CoNLL and MasakhaNER datasets. Our experiments don't include German, thus it is not included in our macro average. mBERT obtains a score of zero on Amharic due to having no vocabulary entries in the Amharic script.

C Additional material for the translation experiment

Input	Ground truth	Canine output
"You saw?" he said.	–Tu as vu? dit-il.	– Vous êtes prises, dit Arthos Conseigne a Marguteries.
"We shall have to beat the forest," said the engineer, "and rid the island of these wretches.	– Il faudra battre la forêt, dit l'ingénieur, et débarrasser l'île de ces misérables.	Ils étaient de la maison, les compagniers, et se regardait avec une chambre explication dans longtemps du souvert l'autres.
Phileas Fogg, having shut the door of his house at half-past eleven, and having put his right foot before his left five hundre	Phileas Fogg avait quitté sa maison de Saville-row à onze heures et demie, et, après avoir placé cinq cent soixante-quinze foi	Il avait été par une fois, et les conseillement de cette heure, il faut pour la plusie dans sous longtemps du bonher se rappel

Table 5: Few inferences for the translation task with the CANINE-BART model

D Extra material for the NLI experiments

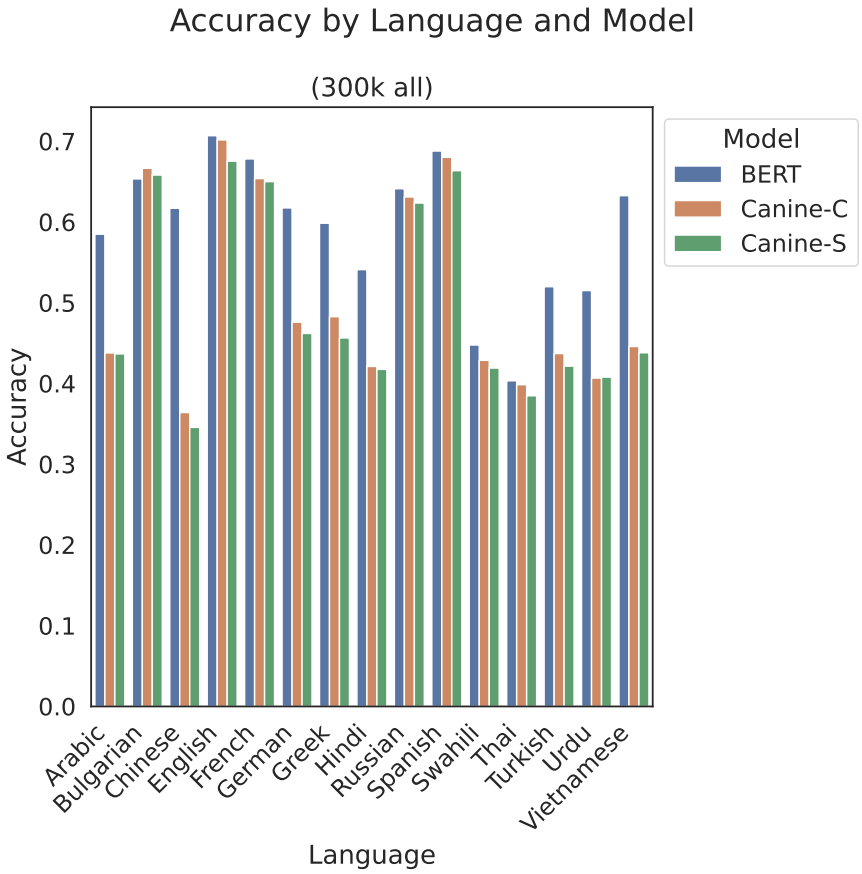

Figure 3: Bar plot of the F1 score of the first experiment, trained on 300k observations containing English, French, Spanish, Bulgarian and Russian, evaluated on the 14 available languages. We see that for the languages in the test set, the performance between the 3 models is similar with a bit of variance. However, when evaluated on languages not seen during training, the results of both CANINE models drop drastically, while BERT's performance is a lot better. For example, when looking at Greek or German, we observe a difference of 15 points, and more than 25 points in Chinese. We suspect that this is due to a better pre training for BERT, but our current experiments are not enough to conclude.

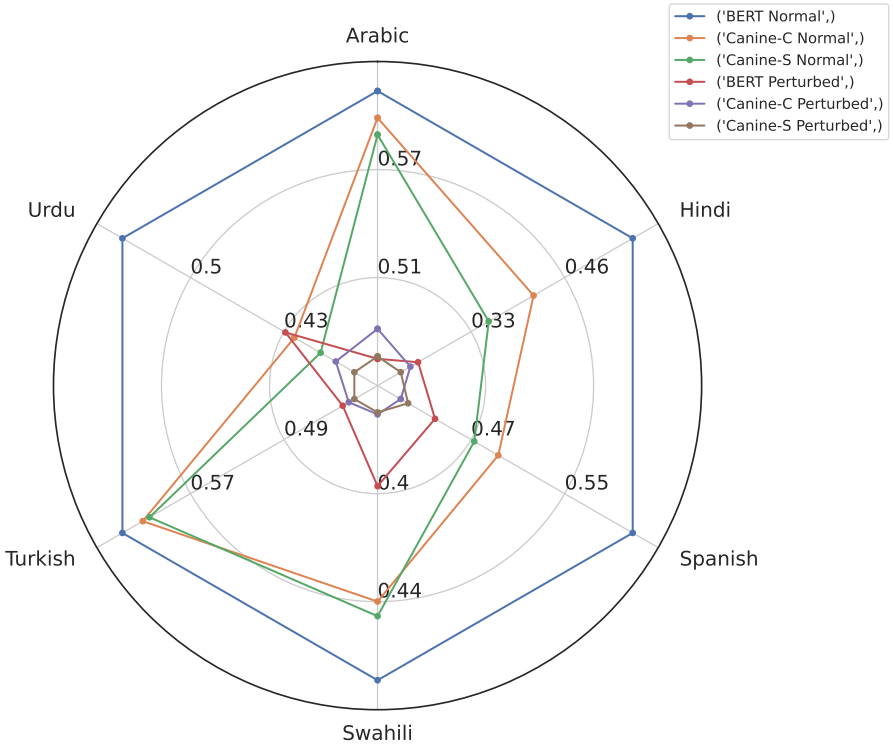

Figure 4: Radar plot of the weighted F1 score for a model trained on 200k of Turkish and Arabic observations. We see that the trained BERT model is always the best in this situation, but both CANINE models are close on the train languages. However, on unseen languages, we see a lot bigger difference.

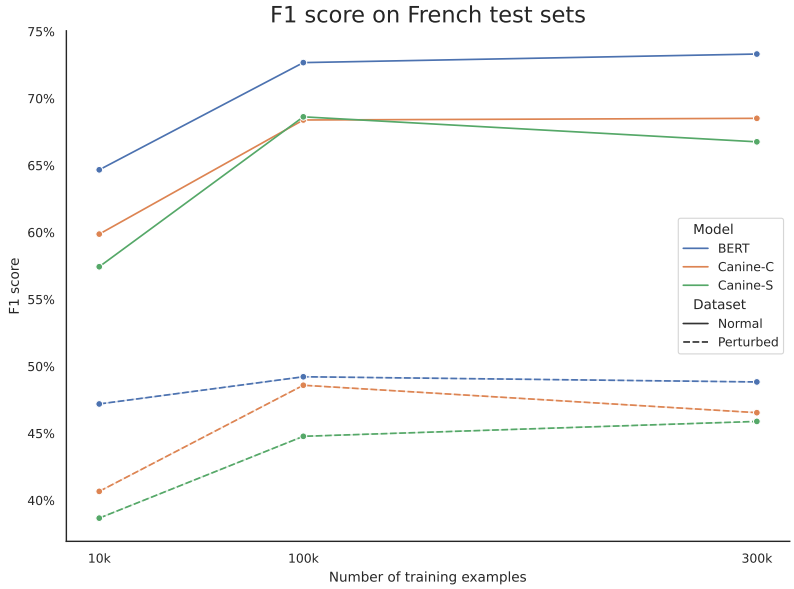

Figure 5: Performance of three models trained on different sized dataset (10k, 100k and 300k) only in French. The plain line is the metrics evaluated on the French test set. While the dotted lines are the metrics evaluated on the perturbed French dataset using *nlpaug*. First, we notice that we have a significant improvement in performance when going from 10k to 100k, but this is not the case from 100k to 300k, this might be due to difference in hyperparameters, training procedure (scheduler and learning rate, etc.) or the number of epoch not being the same (they are still more steps in the 300k). We see that BERT is systematically the best model by quite a margin (>5%). It is more surprising that this stays true for the perturbed test set, even if the gap with CANINE-C is reduced.

	Premise	Hypothesis	Label	BERT predictions	CANINE-C predictions
Original	Eh bien, je ne pensais même pas à cela, mais j'étais si frustré, et j'ai fini par lui reparler.	Je ne lui ai pas parlé de nouveau	Contradiction [2]	(**0.47**, 0.44, 0.09)	(**0.91**, 0.04, 0.05)
Augmented	Eh b ien, je ne p ensa mêm pas à c ela, m ais j ' étai si us tré, et j ' ai fni par lui rep aer.	Je ne lui ai pas palé de oveau	Contradiction [2]	(0.28, **0.63**, 0.09)	(**0.57**, 0.35, 0.08)
Original	Mercredi, Clinton a choisi de parler d'une industrie différente.	Clinton a parlé ce Mercredi.	Entailment [0]	(0.05, 0.1 , **0.84**)	(**0.70**, 0.17, 0.13)
Augmented	Mercredi, Clinton a cih osi de ap rlre d ' une indu strie dfiéfrente.	Clniotn a paré ce recdei.	Entailment [0]	(**0.50**, 0.05, 0.45)	(**0.51**, 0.08, 0.41)
Original	Et maintenant j'ai une sœur en Allemagne	J'ai une sœur qui parle allemand.	Neutral [1]	(**0.58**, 0.19, 0.22)	(**0.90**, 0.04, 0.05)
Augmented	et mainttan j ' ai une sœr en Allemagne	J ' ai une sœr qui ap rle ea madn.	Neutral [1]	(0.25, 0.14, **0.61**)	(0.19, **0.49**, 0.31)

Table 6: Three examples of prediction for the NLI task, on both original and perturbed inputs, for the French test dataset. The **bold value** represents the value chosen by the model. The underlined value represent the ground truth. We didn't include CANINE-S for the sake of readability, since the results were mostly in line with CANINE-C. We chose one examples for each class randomly. We note that for the first example, both BERT and CANINE-C get it confidently wrong. We see that the perturbation has a bigger impact of BERT's predictions, which is what we would expect, as CANINE-C can somewhat make abstraction of missing characters or extra spaces. The second example also shows that CANINE-C is more robust, it predicts write both times. Lastly, the third example is similar to the first one in the interpretation. Overall, we see that the perturbations reduces the confidences in almost all cases, and impacts BERT more than CANINE-C.