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

Yujin Cho
ENS Paris-Saclay
yujin.cho@ens-paris-saclay.fr

Chloé Sekkat
ENSAE Paris
chloe.sekkat@ensae.fr

Gabriel Baker
Telecom Paris
gabriel.baker@telecom-paris.fr

Loubna Ben Allal
ENS Paris-Saclay
loubna.ben_allal@ens-paris-saclay.fr

Abstract

This project is part of the Algorithms for Speech and Natural Language Processing course of the MVA master. We worked on a tokenization-free encoder for language representation called CANINE [3]. It is the first character-level pre-trained deep encoder that is completely tokenization-free and vocabulary-free. We will study the performance of CANINE on several NLP tasks and compare it to BERT and/or mBERT. Our code is available on our Github [1].

1. Problem Definition

Transformers have revolutionized the field of NLP, with remarkable generalization through transfer learning. However, most of these systems require an explicit tokenization step, which poses many limitations. First, subword tokenizers struggle with noisy text [9], and their method of splitting can also be unsuitable for certain morphologies of languages. Moreover, tokenizers use a fixed vocabulary during pre-training, and this limits their generalization capacity for downstream tasks, especially in specialized domains [2]. These limitations question the need for tokenization. Character-level models, that directly operate on characters instead of tokens, have emerged a while ago, but their performance was poor compared to their word-level counterparts. CANINE [3] is the first pre-trained tokenization-free encoder that outperforms BERT on some specific tasks. It operates directly on character sequences, and uses a downsampling strategy to reduce the input sequence length, followed by a deep transformer encoder. In this project, we studied the performance of CANINE in comparison with BERT for four NLP tasks: Question Answering, Named Entity Recognition, Sentiment Analysis and Natural Language Inference on medical data, we also evaluated the robustness of CANINE to noisy data.

[1] https://github.com/chloeskt/canine_mva

2. Question Answering

2.1. Reproducing paper's results

The first step of our study was to reproduce the paper's results. The authors show the performances of CANINE against mBERT (their own trained version from scratch) on both Passage Selection Task and Minimal Answer Span Task, on TyDI QA dataset, a dataset of information-seeking questions in 11 typologically diverse languages. The difference in hardware (hence in hyperparameters, see Table 4) might explain the discrepancy between our results (Table 5) and theirs. We used the provided source code [2] but had to modified it to make it work correctly.

	SelectP		**MinSpan**	
F1-score	Ours	Papers	Ours	Papers
CANINE-c	56.7	65.7	42.5	53.0

Table 1. F1-score obtained on two evaluation tasks on TyDiQA dataset: Select Passage and Select Minimal Span. Note that our F1-scores are averaged over all languages except English; while the scores reported in the paper might contained English as it is not specified.

2.2. Zero-shot transfer on multilingual data

Next we evaluated CANINE [3] in a different setting (not in the paper): finetuning on a unilingual dataset (SQuADv2 [5]) and then tested in zero-shot transfer setting on multilingual data (XQuAD [1]). We used the HuggingFace (HF) implementation and added one linear layer of shape (768,2) to predict start and end logits for minimal span answer (given a context and a question, extract the passage with the answer in the context). The trickiest part was to implement a functional tokenizer for CANINE since the one provided by HG is not suited for QA. To ensure fair comparison, we finetuned BERT, mBERT and XLM-RoBERTa using pretrained models from HF (BERT-BASE-

[2] https://github.com/google-research/language/tree/master/language/canine/tydiqa

UNCASED, BERT-BASE-MULTILINGUAL-CASED and XLM-ROBERTA-BASE) (see Table 6 for hyperparameters). All models were trained using AdamW optimizer and early stopping on validation set. Table 2 summarizes the best F1-score we were able to get (on created test set). CANINE-C is comparable to other BERT-like models in such settings.

	CANINE-C	CANINE-S	mBERT-base	BERT-base	XLM-RoBERTa
F1-score	**74,1**	72,5	77,51	76,02	**78,3**
EM score	69,2	69,6	74,1	73,08	**75,12**

Table 2. F1 and Exact Match scores obtained on SQuADv2

Then we evaluated the models in zero-shot transfer setting on 11 other languages with various morphologies. We used XQuAD dataset [1] (Table 7) (Arabic, German, Greek, Spanish, Hindi, Russian, Thai, Turkish, Vietnamese, Chinese and Romanian). Results can be found in Tables 8 & 9. BERT performs poorly since it was only pretrained on English. CANINE-C is not great (-10F1 w.r.t. BERT, -20 w.r.t. XLM-RoBERTa) to the exception that it performs better than mBERT in Thai (isolating language; +10F1). The gap between XLM-RoBERTa and CANINE-C increases when evaluated on languages such as Vietnamese, Thai or Chinese. These languages are mostly isolating ones i.e. language with a morpheme per word ratio close to one and almost no inflectional morphology.

2.3. Robustness to noise

This experiment is meant to test the abilities of CANINE to handle noisy inputs, especially noisy questions as in real life settings the questions are often noisy (misspellings, wrong grammar, etc - think of ASR systems or keyboard error while typing). We created 3 noisy versions of SQuADv2 test set: with probability p, each word, in the question, gets transformed into a misspelled version of it (substitution of characters). Results (Table 10) hints that once the noise level is high (i.e. $> 40\%$), both CANINE-C and CANINE-S perform similarly to BERT-like models. CANINE-S is even better than mBERT and BERT.

2.4. Discussion

In our zero-shot transfer QA experiments, CANINE does not appear to perform as well as token-based transformers such as mBERT. They are able to learn cross-lingual embeddings for given tokens even if finetuned only on English (analytical language). Whereas CANINE cannot adapt well in zero-shot transfer to isolating languages (Thai, Chinese) and synthetic ones with agglutinative morphology (Turkish) or non-concatenative (Arabic) in this setting. But it works decently well for languages close enough to English, e.g. Spanish or German. While mBERT and CANINE have both been pretrained on the top 104 languages with the largest Wikipedia using a MLM objective, XLM-RoBERTa was pretrained on 2.5TB of filtered CommonCrawl data containing 100 languages. This might be a confounding variable. Finally, it seems that when artificial noise levels are high, CANINE-S is preferable to BERT as it is fairly robust to this type of noise.

3. Named Entity Recognition

The main task of theses datasets consists on language-independent named entity recognition. It focuses on four types of named entities: persons, locations, organizations and names of miscellaneous entities that do not belong to the previous three groups. We propose 3 experiment, in order to evaluate CANINE model on NER task. We will compare and test the robustness of CANINE with BERT model.

3.1. Dataset

We will use CoNLL-2002 [10] and CoNLL-2003 [11]. This dataset is composed of different languages corpus mainly coming from newspaper article.

Each corpus is divided into train, validation and test set with manually annotated label (Table 11 in Appendix). Each word of the corpus will have a label corresponding to his meaning (Table 12 in Appendix). The B-XXX tag will be used to tag respectively the first element of a named entity and the I-XXX tag is used for all other words in named entities of type XXX (i.e. "Yujin CHO" will have label **(1)B-PER** then **(2)I-PER** because it belongs to the same named entity of type PER).

3.2. Architecture

The NER task can be considered as a classification problem, where each token will have to be assigned to a class number. For this we use a CANINEFORTOKENCLASSIFICATION from the transformer library of HuggingFace. This model consists of a linear layer on top of the hidden-states output of CANINE model. The final will have a size of 9 as there is 9 different classes to predict (Table 12 in Appendix).
The most tricky part was to build the data pipeline as ConLL dataset from HuggingFace was pre-processed for word tokenizer architecture. First step was build back a sentence from word tokens by adding a space token between each word, some special punctuation characters had special handling rules (i.e. no space before a coma, ...). We had to generate new label specific to CANINE architecture, goal is to propagate the label to all characters of a word to compute well loss on each character prediction. Finally we had to process the output of CANINE model to select the majority class value among all letters from a word to predict the final label of a complete word. The final finetunings are the followings, the model was trained during 4 epochs, with a learning rate of $2e^{-5}$ and weight decay of $1e^{-5}$, AdamW for optimizer and Cross Entropy loss.

3.3. Pre-Training method

CANINE model was provided with 2 methods of training, CANINE-C is pre-trained with autoregressive character loss and CANINE-S is pre-trained with subword loss. Both of this training methods will differ the features learnt by the model.

Our goal is to see impact of this different training on NER task.

Model	Spanish	English	Dutch
CANINE-C	94.6	91.4	95.1
CANINE-S	94.6	91.3	94.5

Thanks to this first experiment, we could see that pre-training doesn't impact on the NER task score, for the rest of our experiment we will continue to use CANINE-C model.

3.4. Transfer learning

Train / Test	Spanish	English	Dutch
Spanish	**94.6**	77.4	86.0
English	91.1	**91.4**	91.1
Dutch	88.6	74.1	**95.1**

From this experiment, we can notice that a model trained with English performs honorably well on other languages, for Dutch, it can be explained by the fact that it is the closest language to English has it possesses many words and phrases similar to English and has a similar grammatical structure. It can also be explained by the fact that English have a world-wide influence in Organisation names. Spanish and Dutch trained model performs well on their own language but performs similarly poorly in other language.

3.5. CANINE vs BERT on NER

For this last experiment, we decided to compare BERT model trained for NER task with dataset ConLL2003(English) with our CANINE-C model trained on the same dataset. Unfortunately, we didn't succeed to reproduce the original paper score, we suspect an issue of tokenization in long sentences as the predictions are similar to original label but shifted from few tokens which result in a wrong F1 score.

Model	Test
CANINE-C (ours)	91.4
BERT (ours)	75.6
BERT (paper)	91.3

Thanks to this final experiment we can compare the score reached by CANINE and compare to BERT as benchmark. The F1 score of both model are quite similar ± 0.1.

3.6. Results and discussion

The method of pre-training doesn't impact the results on NER task. A big dataset helps the model to generalize in other language. CANINE particularity of character tokenization don't bring revolution in NER task. The main advantage is the size of CANINE model that has $\sim$ 3 times less parameters than BERT (121M vs 340M). The CANINE model trained on English dataset generalize well on other language tested without supplementary training.

4. Natural Language Inference on Medical data

For this task, we are interested in Natural Language Inference (NLI). We are given two sentences, a hypothesis and a premise, and we have to predict if the hypothesis is true, false or neutral given the premise. These relations correspond respectively to the labels Entailement, Contraction and Neutral. We want to perform this task on MedNLI [6], a medical dataset annotated by doctors for NLI. Statistics of the data can be found in Table 13.

The choice of this task, is motivated by the fact that CANINE doesn't use a tokenizer, that can hinder the generalization capabilities of models like BERT to specific domains like medicine. We conducted several experiments to evaluate the performance of CANINE on this dataset and compare it to BERT. Another motivation for this task is related to noise robustness, since CANINE doesn't use a fixed tokenization vocabulary, which would be insufficient for noisy text with lots of new words, we expect CANINE to perform better than BERT on noisy data. We tested this hypothesis by adding noise to the MedNLI dataset and evaluating the performance of the model for different noise levels.

4.1. BERT on MEDNLI

We fine-tuned a pretrained BERT on MedNLI. We used the "base-uncased" version of BERT available in the Hugging Face Hub. The model has 110M parameters, uses 12 Transformer layers with 12 attention heads and 768 as a hidden dimension. As for the hyperparameters, we used a batch size 16 of and a maximum sentence length of 256 both both sentences. We used Adam optimizer with a learning rate of 3e-5 and a waight decay of 0.01 and a linear scheduler with warmup for 0.1 of total steps. We were able to get a test accuracy of 77.6 ± 0.6.

4.2. CANINE on MEDNLI

We fine-tuned CANINE on MedNLI, we tested CANINE-S with subword loss and CANINE-C with autoregressive character loss, we got slightly better results with the latter. Two important parameters were the batch size and the maximum length of sentences, corresponding to the number of characters allowed. Due to memory issues, we had to find a trade-off between these two parameters. We found that only few premise sentences have more than 600 characters and few hypothesis sentences have more than 100 characters, so we used 700 as the maximum length for the concatenation of both sentences. As for the batch size, we were able to use a maximum value of 16. We used the same optimization scheme as for Bert fine-tuning. This model has 127M parameters and the training takes more time in comparison with BERT. For three different seeds, we got a test accuracy of 73.07 ± 0.3.

Model	Test accuracy
BERT	77.6
CANINE	73.07

CANINE accuracy is less than BERT's. This can be explained by the memory trade-off we had to make to increase batch size and maximum length, or BERT may probably be not as limited in generalization to medical domain as we assumed since it was trained on very large and various data.

4.3. Robustness to noise

To evaluate the model's robustness to noise. We created noisy versions of the MEDNLI corpus where, given a noise level p, we transform each token with the probability p into a misspelled version either by removing, adding, replacing a single character or swapping two consecutive characters. We conducted experiments where noise is added to the test set only as well as experiments where it is added to the training and validation sets as well.

We first tested our fine-tuned models on clean MedNLI data on three noisy test sets, corresponding to three noise levels: 10%, 20% and 40%. The left plot in figure 1 shows CANINE and BERT Performance for each level. We can see that CANINE is more robust than BERT for noise levels 20 and 40, and the difference is very significant for noise level 40. This suggests that CANINE can be very suitable to noisy text, from social media for instance, where there are many typos and new words that constitute out-of-vocabulary words in BERT tokenization.

We wanted to see if the models can learn to generalize better when trained on noisy data as well, so we added noise to the training and validation sets as well. We can see in right plot in figure 1 that indeed the models generalize better, but for high noise levels such as 40%, CANINE is preferred to BERT.

5. Sentiment Analysis

The last task in which we experimented with CANINE was Sentiment Analysis, which, in its most basic form, consists of predicting if a given text has positive or negative opinions of its subject.

For this task we performed two experiments, one to compare our model with a baseline and the other to test the strengths and limits of CANINE, a different dataset was used for each of them. For the first objective, we opted for SST-2 [8], part of the GLUE benchmark [12], because its is widely used by the NLP community and it is thus a good candidate to evaluate and compare the performance of a model. For the second objective we opted for Sentiment140 [4], which contains 1.6 million tweets, this choice was due to the fact that it contains a language register more prone to abbreviations and colloquialisms, in which CANINE has a theoretical advantage.

5.1. SST-2 experiment

Because SST-2 doesn't have publicly available labels for its test split, we decided to use the validation as test and 5% of the train as validation. The number of samples per split can be seen in table 14

We then used the CanineForSequenceClassification model from HF, which adds a classification head on top of the CANINE pooler, containing a linear layer of 768 neurons, Tanh activation and dropout of 0.1, followed by a final linear layer of 2 neurons. We finetuned the pretrained CANINE-S model for 5 epochs, using AdamW with a weight decay of $1e^{-2}$ and a linear learning rate schedule with warmup for the first 10% of the steps.

To compare our results we used DistilBERT [7], since it has a similar performance to BERT, but it's lighter and faster to train. DistilBertForSequenceClassification from HF was used with the same training hyperparameters as CANINE.

Instead of simply comparing the two architectures, we thought it would be interesting to test the robustness to noise as it was previously done, in order to see if the same tendencies would appear in a different task. For this, we randomly substituted a letter by another random one with different probability levels: 0%, 10%, 20% and 40%. This noise is added both during test and training.

As we can see in the figure 2, CANINE has a worse accuracy with a clean dataset, but this changes as we increasingly add noise, and for the last two noise levels CANINE has a better accuracy. We can also compare our results with the DistilBERT paper, which achieved 91.3% accuracy without noise, close to the 90.9% achieved by us with the same architecture and considerably better than the 86.9% achieved by CANINE.

5.2. Sentiment140 experiment

The Sentiment140 dataset doesn't contain a validation set and, for this reason, we used 1% of the training set, or 16k samples, as validation. We also decided to remove neutral sentiment examples from the test because they are not present in the training. The final dataset statistics can be seen in the table 15.

The experimental setting regarding the networks and hyperparameters used is the same as the one of the previous subsection, but we only tested with clean data as this dataset is too big to execute multiple trainings to evaluate noise robustness, furthermore, given the dataset nature, it already contains inherent noise. The results of CANINE are comparable with DistilBERT, 84.12% and 85.52% respectively. Although CANINE has a slightly lower accuracy, the gap between the models is much lower than with the previous dataset, showing its robustness to artificial and real noise. The table 3 shows the accuracies in all splits for both models.

	Train	**Validation**	**Test**
CANINE-S	91.94%	87.41%	84.12%
DistilBERT	92.25%	87.07%	85.52%

Table 3. Accuracies of CANINE and DistilBERT in Sentiment140 dataset.

6. Conclusion

In general CANINE performed worse than BERT, but, in the 3 tasks that this happened, the outcome was reversed after heavily adding noise to the data. There were also cases where both performed similarly, in the task of NER and when using twitter data. The better relative results of CANINE in these 2 cases can probably be explained by the fact that the word structure is important for the former and that the latter has a

more informal language, both of which benefit from character level embedding.

7. Contributions

Each student worked on different evaluation tasks. C. Sekkat worked on Question Answering, G. Baker worked on Sentiment Analysis, Y. Cho worked on Named Entity Recognition, and L. Ben Allal worked on Medical Natural Language Inference.

References

[1] M. Artetxe, S. Ruder, and D. Yogatama. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856, 2019.

[2] H. E. Boukkouri, O. Ferret, T. Lavergne, H. Noji, P. Zweigenbaum, and J. Tsujii. Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters. *arXiv preprint arXiv:2010.10392*, 2020.

[3] J. H. Clark, D. Garrette, I. Turc, and J. Wieting. CANINE: Pre-training an efficient tokenization-free encoder for language representation, 2021.

[4] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250, 2016.

[6] A. Romanov and C. Shivade. Lessons from natural language inference in the clinical domain.

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[8] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[9] L. Sun, K. Hashimoto, W. Yin, A. Asai, J. Li, P. Yu, and C. Xiong. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985*, 2020.

[10] E. F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002.

[11] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[12] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Appendix

A. Question Answering

A.1. Reproducing paper's results

	Ours	Papers
Maximum sequence length	2048	2048
Doc stride	512	512
Maximum question length	256	256
Batch size	**4**	512
Learning Rate	5e-5	5e-5
Nb of epochs	**5**	10
Warmup proportion	0.1	0.1
Compute system	**Tesla P100 (16Gb)**	Multiple TPUs

Table 4. Training parameters and configuration on TyDI QA experiments.

	SelectP		MinSpan	
F1-score	Ours	Papers	Ours	Papers
CANINE-c	56.7	65.7	42.5	53.0

Table 5. F1-score obtained on two evaluation tasks on TyDiQA dataset: Select Passage and Select Minimal Span. Note that our F1-scores are averaged over all languages except English; while the scores reported in the paper might contained English as it is not specified.

A.2. Zero-shot transfer on multilingual data

	BERT	mBERT	XLM-RoBERTa	CANINE-C	CANINE-S
Batch size	6	6	6	4	4
Learning Rate	3e-5	3e-5	3e-5	5e-5	5e-5
Weigh decay	0	0	0	0.01	0.1
Nb of epochs	2	4	4	2	6
Number of training examples	132335	132335	131823	130303	130303
Number of validation examples	12245	12245	12165	11861	11861
Max sequence length	348	348	348	2048	2048
Doc stride	128	128	128	512	512
Max answer length	30	30	30	256	256

Table 6. Hyperparameters choice for each model. Training was done on one Tesla P100, 16GB of VRAM; dataset used is SQuADv2.

Nb of samples	Training	Validation	Test
SQuADv2	130 319	10 686	1 187
SQuADv1.1	87 599	10 570	-
XQuAD (per language)	-	1 190	-

Table 7. Number of samples per splits and datasets.

	CANINE-C	CANINE-S	mBERT-base	BERT-base	XLM-RoBERTa
English	78,77	79,03	83,59	82,3	**82,8**
Arabic	43,78	29,74	54,09	11,76	**62,48**
German	59,57	55,35	68,4	19,41	**72,47**
Greek	46,93	30,82	56,47	10,21	**70,93**
Spanish	60,47	59,48	72,84	19,72	**75,18**
Hindi	35,21	30,93	51,06	11,07	**62,1**
Russian	60,49	55,09	68,33	9,47	**73,12**
Thai	**37,28**	31,2	**27,63**	10,04	**65,21**
Turkish	31,09	23,83	44,62	16,76	**65,34**
Vietmanese	43,14	35,52	64,49	24,63	**73,44**
Chinese	34,86	28,68	52,71	8,15	**65,68**
Romanian	56,62	43,69	69,31	20,03	**74,78**
Average	**49,02**	41,95	59,46	20,30	**69,16**

Table 8. F1-scores on XQUAD dataset in zero-shot transfer setting. Results are averaged over 3 runs.

	CANINE-C	CANINE-S	mBERT-base	BERT-base	XLM-RoBERTa
English	67,38	66,34	79,51	69,57	**72,18**
Arabic	26,25	13,75	37,22	4	**45,79**
German	43,16	38,27	50,84	4,9	**55,21**
Greek	29,14	13,42	40,16	5,37	**53,19**
Spanish	42,74	39,57	54,45	4,7	**56,3**
Hindi	18,93	16,54	36,97	4,8	**45,042**
Russian	43,48	35,65	52,1	4,62	**55,54**
Thai	20,5	17,91	21,26	2,6	**54,28**
Turkish	14,8	10,11	29,41	4,87	**48,85**
Vietmanese	25,17	19,65	45,21	7,64	**54,02**
Chinese	21,36	20,2	42,26	3,1	**55,63**
Romanian	39,98	26,5	54,62	6,21	**61,26**
Average	**32,74**	26,49	45,33	10,20	**53,19**

Table 9. Exact-Match scores on XQUAD dataset in zero-shot transfer setting. Results are averaged over 3 runs.

A.3. Robustness to noise

	Noise level 10%		Noise level 20%		Noise level 40%	
	F1 score	**EM**	**F1 score**	**EM**	**F1 score**	**EM**
BERT	73,68	70,79	71,22	68,55	66,42	63,74
mBERT	74	70,75	71,66	68,46	67,08	64,74
XLM-RoBERTa	**74,54**	71,61	**72,68**	69,81	67,12	64,43
CANINE-C	69,64	66,89	67,88	65,43	66,03	63,9
CANINE-S	72,25	69,65	70,3	68,03	**67,18**	64,6

Table 10. F1 and EM scores reported on three noisy test sets extracted from SQuADv2 (averaged over 3 runs).

B. Name Entity Recognition

B.1. Dataset

Language	Train	Validation	Test
English	14042	3251	3454
Spanish	8324	1916	1518
Dutch	15807	2896	5196

Table 11. Number of samples for ConLL2002 and 2003 dataset.

Label	Name	Meaning
0	O	nothing
1	B-PER	Person (B)
2	I-PER	Person (I)
3	B-ORG	Organisation (B)
4	I-ORG	Organisation (I)
5	B-LOC	Localisation (B)
6	I-LOC	Localisation (I)
7	B-MISC	miscellaneous (B)
8	I-MISC	miscellaneous (I)

Table 12. Label description for ConLL2002 and 2003 dataset.

C. Natural Language Inference on Medical data

C.1. Dataset

	Train	Validation	Test
# samples	11232	1395	1422

Table 13. Number of samples per split in MedNLI dataset

C.2. Robustness to noise

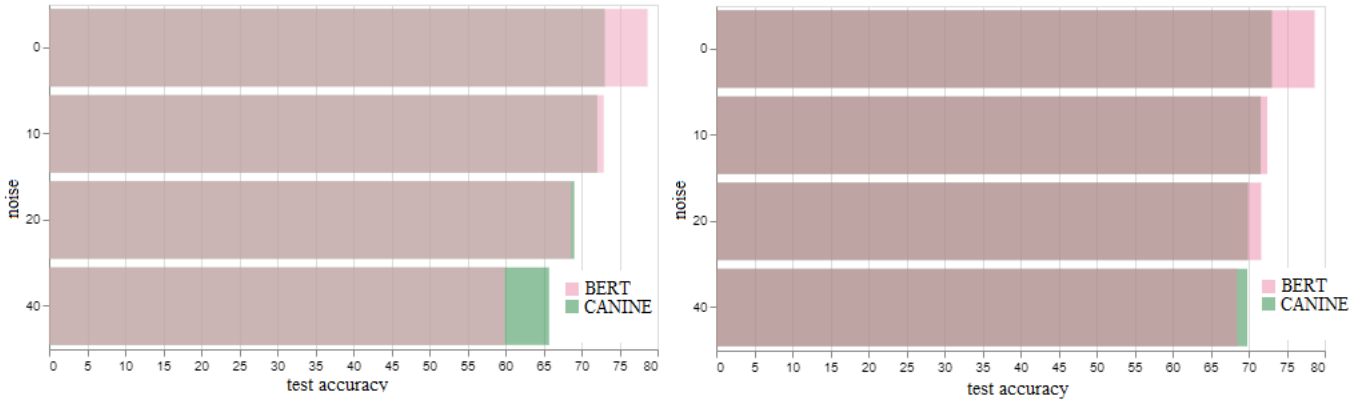

Figure 1. MedNLI test accuracy per noise level. The LEFT plot corresponds to testing a model that was fine-tuned on clean data. In the right plot, the models were trained on noisy data with the same noise level as the test set.

D. Sentiment Analysis

D.1. SST-2 experiment

	Train	Validation	Test
# samples	63981	3368	872

Table 14. Number of samples per split in SST-2 dataset.

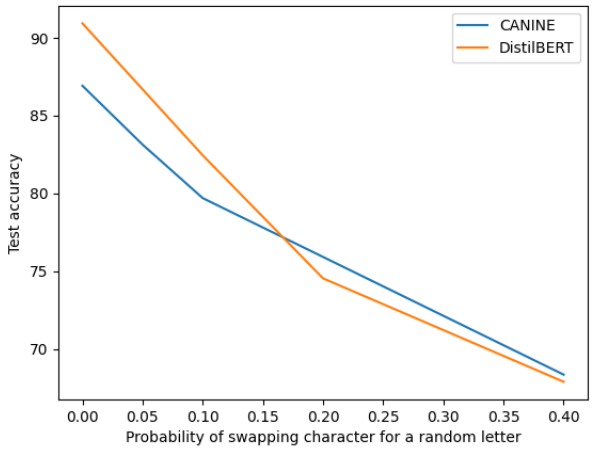

Figure 2. Accuracy per noise level for CANINE and DistilBERT.

D.2. Sentiment140 experiment

	Train	Validation	Test
# samples	1440000	16000	359

Table 15. Number of samples per split in Sentiment140 dataset.