Project Report : CANINE

Thomas Palmeira Ferraz thomas.palmeira@telecom-paris.fr
Loïc Magne loic.magne@outlook.com
Marah Gamdou marah.gamdou@student-cs.fr
Marius Roger marius.roger@outlook.fr

Abstract

In this project, we demonstrate when it is worth using CANINE by comparing it with BERT on text classification. In addition, we verify to what extent character-level embedding does not depend on the existence of clean and well-spelled data as mentioned in the CANINE paper. Finally, we evaluate whether CANINE exhibits more robustness than the traditional BERT to verify the claims that the character-level language model exhibits greater robustness to adversarial manipulation.

1. Introduction

Several encoders for language representation require an explicit tokenization step, which has many drawbacks such as the sophistication it requires and the impossibility of writing rules for all existing languages and domains. Furthermore, tokenization-based models are generally sensitive to input corruptions, whether from natural typos (noise in the data) or from adverse manipulation. These last two phenomena being important and especially recently, CANINE has appeared as a potential solution which promises to solve these and other problems thanks to its mode of operation which does not need tokenization.

In this project, we will verify its ability to solve the above-mentioned problems while checking that it provides an acceptable performance for classical NLP tasks like text classification.

We will therefore compare CANINE to Bert in three experiments to verify the points mentioned above.

2. Reproducing paper results

Marius Roger was not able to reproduce the results from the paper for either CANINE-C or CANINE-S due to a systematic colab stop signal while preparing the TyDi QA data needed to fine-tune the model (and test it) and not being able to run the code on his own computer.

3. Evaluation on text classification

Since it is important to compare CANINE with BERT on other NLP tasks that were not tested in the paper in order to see to what extent the conclusions of the authors are valid, we will compare both models on text classification task using IMDB dataset [1]. This dataset is a Large Movie Review Dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. It contains 25,000 movie reviews for training and 25,000 for testing. The dataset contains two classes: positive and negative reviews. Some samples are presented in the table 1. The training set and the test set are balanced (50% positive reviews and 50% negative reviews).

We trained both CANINE and distilbert (as it is lightweight) on text classification using the latter mentioned dataset with various training configurations. The table 2 shows the obtained performances.

Notice that in the table we have three types of experiments :

- We used already fine-tuned models (both distilbert and canine) directly without training
- We fine-tuned both models starting from pre-trained models (canine was not totally pre-trained due to memory limitations however most of its layers were fine-tuned)
- We only fine-tuned the classifier for both models

Notice that for all the experiments mentioned above (the different trainings) we used the same hyperparameters that are usually used by hugging face in such scenarios without trying various hyperparameters for each case. The leaning rate=$2*10^{-5}$, the train bach size=16, the weight decay=0.01 and the number of epochs=5.

Note that in all (training) scenarios , DistilBert outperforms CANINE. This can be due the fact that Bert-like models are more generalizable to new tasks than CANINE. In fact, even if we are using DistilBert which is a small, fast, cheap and light Transformer model trained by distilling BERT base, we managed to have better performance

[1] https://huggingface.co/datasets/imdb

Sample	Class
In New York, when the shy and lonely project manager of a design firm Matt Saunders (Luke Wilson) meets Jenny Johnson (Uma Thurman) in the subway, he invites her to date and have dinner with him. Jenny immediately falls in love for him, they have sex and she discloses her true identity to him...	Positive
This is, quite literally, the worst movie I have ever watched in my life. It may be the worst movie possible. Some movies are so bad that they're good; this movie is so bad that it goes past enjoyable camp and simply becomes unwatchably awful. It is the anti-enantiodromia...	Negative
It makes the actors in Hollyoaks look like the Royal Shakespeare Company. This movie is jaw dropping in how appalling it is. Turning the DVD player off was not a sufficient course of action. I want to find the people responsible for this disaster and slap them around the face. I will never get that time back...	Negative

Table 1. Samples from IMDB Dataset

Training	Test Accuracy	Test Precision	Test Recall
canine-s without finetuning	0.500	0.500	0.999
distilbert-base-uncased-finetuned-sst-2-english without finetuning	0.828	0.848	0.799
canine-s-finetuned-sst2 without finetuning	0.745	0.846	0.599
distilbert-base-uncased fully finetuned	**0.867**	**0.860**	**0.877**
canine-s mostly finetuned (not the whole network due to memory limitations)	0.576	0.560	0.716
distilbert-base-uncased only the classifier is finetuned	0.846	0.846	0.856
canine-s only the classifier is finetuned	0.527	0.547	0.312

Table 2. Text classification on IMDB Dataset

than CANINE (which was more computationally demanding for this experiment).

This part of the project was done by Marah Gamdou.

4. Noisy data

As mentioned in the CANINE paper, using character-level embedding has several advantages: it requires less preprocessing engineering on the input data, doesn't need for each token to appear often, etc. Among them, a particularly interesting one is that character-level embedding doesn't rely on having clean well spelled data. In those experiments, we try to leverage this fact to evaluate the effect of character-level embedding versus subword embedding models like BERT. The goal is to compare the benchmark with CANINE on a dataset with high variance, where words can be misspelled in several ways, with slang language which can be out of the distribution of the training corpus. To this end, we used a Twitter hate speech detection dataset[2], found on HuggingFace. This dataset is a typical sentence classification task, where each sentence is tweet, which must be classified as either 'hate-speech', 'offensive-language' or 'neither'. The dataset contains 24783 samples, with imbalanced class repartition: 1430 (5.8%) 'hate-speech' tweets, 19190 (77.4%) 'offensive-language' tweets, and 4163 (16.8%) 'neither' tweets. To account for that, we use metrics like balanced accuracy to assess performances. The dataset is split into 90%, 10% training/test set with the same class repartition. Table 3 shows some samples of the dataset with their labels.

We run experiments for BERT and CANINE in two different setups:

- Training the full models, starting from a pre-trained model
- Training only a classification layer above models, starting from a pre-trained model

Figure 1 show balanced-accuracy performances on each scenario for each models.

Overall, only finetuning the last layer seems to provide better results. When fine-tuning the full network, CANINE performs slightly better than BERT. However when only fine-tuning the last layer, BERT seems to perform significantly better than CANINE. This is the opposite than what we would have hoped: ideally CANINE would only need a final classifier to work well on noisy data, since it's supposed to be less sensitive to noise. Nevertheless the results obtained must be taken with care, here are some critical thoughts of the experiments:

- Due to resources constraints, each setups could be trained only once without hyper parameters tuning, for a low amount of epochs (1 epoch took around 30 minutes to train with available resources)
- For the same reason, only one dataset was considered, which is not enough to draw conclusion on the performances on noisy data.

[2] https://huggingface.co/datasets/tweets_hate_speech_detection

[2] https://huggingface.co/datasets/hate_speech_offensive

Sample	Class
@mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!	offensive-language
trust me, just set the alpha to 0. it's just as good (maybe better) than setting the visibility to false. now: fly on silver bird!	neither
@HeauxmerSimpson I'm jus tryna vaca away from the niggers bro	hate-speech
@ChaseBasford chase you God damn faggot	hate-speech

Table 3. Samples from tweets hate speech dataset

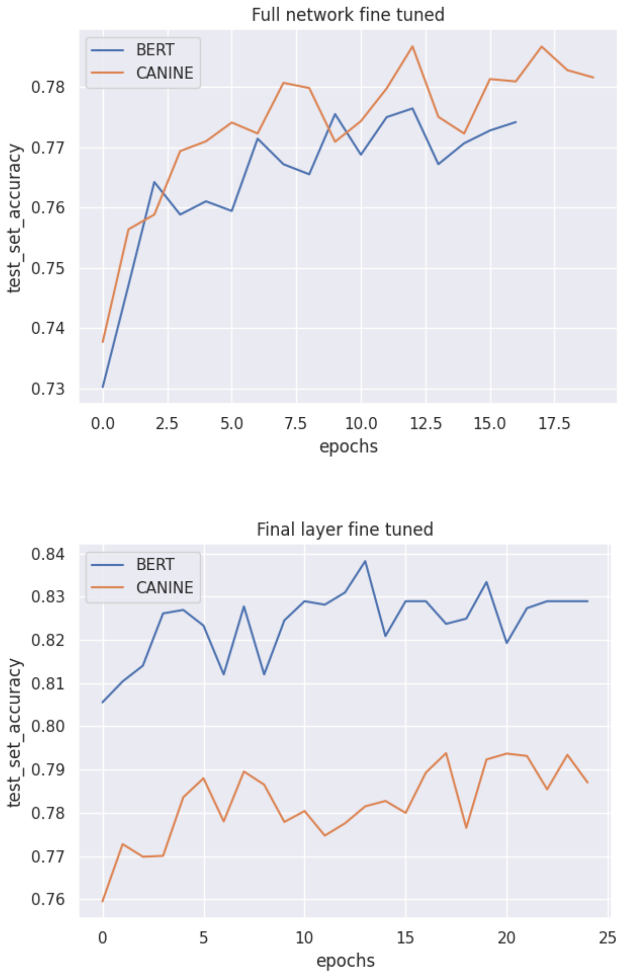

Figure 1. Balanced accuracy on hate speech classification task, (a) full network finetuned (b) final classifier layer finetuned

This part of the project was done by Loïc Magne.

5. Adversarial Robustness

Adversarial Attacks are defined as small and human-indistinguishable changes in an input that lead to large changes in the output in a model, so as to fool it. Adversarial Robustness is the model's ability to resist being fooled, ie., its ability to defend itself against adversarial attacks. In the context of NLP, attacks refer to changes in a text that preserve its original meaning for humans, but can deceive models. These attacks can be of three levels: character-level (changes in characters e.g. random swaps, deletions, and additions, which look like typos to humans), word-level (replacing words keeping the meaning of the sentence, usually replacing them with synonyms), and sentence-level (generating paraphrases).

CANINE [1] claims that the tokenization-free encoder approach adopted is better able to resist adversarial manipulations and typos than the WordPiece tokenization [6] approach adopted by the classic BERT [2]. However, the cited work did not present practical evidence to support this claim. Here in this work we perform a preliminary experiment to contradict or support this hypothesis raised by the authors.

The experiment consists of training models on a part of a splitted dataset, and evaluating it on an unseen part in two ways: first directly evaluating the dataset, then using the same dataset to generate attacks on the model using the Pruthi Algorithm [5]. Finally, the performance metrics from both tests are compared in order to see how much the model has lost performance due to the attacks.

Pruthi's algorithm [5] consists of generating adversary examples that are able to disrupt sentences with four types of character-level edits: (1) Swap: swapping two adjacent internal characters of a word. (2) Drop: removing an internal character of a word. (3) Keyboard: substituting an internal character with adjacent characters of QWERTY keyboard (4) Add: inserting a new character internally in a word. To maintain human-readability (and consequent indistinguishability), the constraints only allow to modify the internal characters of a word, and not edit stopwords or words shorter than 4 characters. For 1-character attack, all combinations possible are tested, while for more than one character greedy search is performed until finding an example that modifies the output. This reduces computational time, but does not deliver an optimal solution, reason why the performance obtained under attacks is only an upper bound.

For this experiment, we use the IMDb Dataset [3], a well-recognized benchmark dataset for sentiment analysis, in which real movie reviews are annotated positive or negative. The train split of the dataset is used for fine-tuning

[2] Dataset avaiable at: https://huggingface.co/datasets/

	CANINE	**BERT**
Original accuracy	84.38%	90.63%
Accuracy under attack	20.63%	10.63%
Attack success rate	75.56%	88.28%
Average perturbed word %	15.41%	15.37%
Avg. num. queries	89.87	82.25

Table 4. Performance comparison between the fine-tuned models CANINE and BERT, when exposed to adversarial attacks on the IMDb dataset.

both BERT and CANINE for 5 epochs, with a learning rate of 10^{-5}, batch size equal to 6 (most the GPU could fit). A Nvidia Tesla T4 15GB GPU is used. Split validation of the dataset is used to choose the best model among the epochs. The split test is then introduced into the model and also used to generate attacks, taking accuracy as a performance comparison metric. We use th he implementation of the algorithm available in the TextAttack library [4]. Table 4 presents the results of the experiments.

The results indeed present an advantage to CANINE. Numerically, the attacker was more successful against BERT than against CANINE (88.28% versus 75.56%). Also, CANINE obtained a higher under attack accuracy than BERT (20.63% against 10.63%), a contrast to the original accuracy in which BERT beat CANINE (90.63% to 84.38%), which demonstrates worthiness in losing a little of performance on standard data to ensure better responses to adversarial cases. CANINE also presented a slight higher resistance to attacks, having on average 89 queries necessary to be fooled, while BERT was fooled on average with 82 queries. The average of words in the dataset used was 42.3.

However, it is worth emphasizing that the numbers presented are upper bonderies, given the greedy nature of the search adopted. Therefore, these are not enough to state that the adoption of CANINE guarantees greater adversarial robustness than BERT. In addition, the experiment is limited since the test was restricted to only one dataset. For a more assertive outcome, on availability of adequate computational resources, it would be necessary to compare the models in different datasets and using different search strategies, including those that present optimal solutions. Even so, the results do not refute the thesis of Clark et al. [1], providing foundations for future investigations.

This part of the project was done by Thomas Palmeira.

6. Conclusions

In these projects we compared CANINE to Bert in different tasks and with different datasets to verify the robustness it is able to provide by not using the tokenization step. Although we do not see a clear advantage to using CANINE over Bert for classical NLP tasks such as text classification or in cases where there is natural noise in the data, we do see an interest in using it in the case of adversarial data.

However, Our study, even if it is based on different experiments and datasets at each time, it can be influenced by our choice of settings in each experiment since we do not re-test for several datasets. It is therefore important to test our work in the future using more datasets to confirm the generalization of our conclusions.

References

[1] 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. 3, 4

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 3

[3] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. 3

[4] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, 2020. 4

[5] Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. Combating Adversarial Misspellings with Robust Word Recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy, July 2019. Association for Computational Linguistics. 3

[6] 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. 3

imdb.