Algorithms for Speech and Natural Language Processing

MVA 2021-2022

David Soto
david.soto.c17@gmail.com

Elías Masquil
eliasmasquil@gmail.com

Nicolás Violante
nviolante96@gmail.com

Introduction

In this project we compare the performance on classification downstream tasks of the character-based language model CANINE against the subword-based mBERT. For both cases, we use the pre-trained models available at the Hugging Face Hub [1] and fine-tune them for the particular downstream task.

Why character-based tokenization?

Subword tokenization is not appropriate for all types of languages. As stated in the paper, this approach is incorrect for languages that heavily rely on compounding (e.g German, Icelandic). It can also be inappropriate to use it in languages with non-concatenative morphology (e.g Arabic, Hebrew). Let's consider the following word in Icelandic[2]:

"heildarraforkuþörf" (total electric energy requirement)

- **Constituent structure:** [heildar, raf, orku, þörf] (total, electric, energy, need)
- **Cased tokenization:** ['he', '##ild', '##arra', '##for', '##ku', '##þ', '##ör', '##f']

Using mBERT subword tokenizer, results in an"arbitrary" tokenization that is derived from statistical properties, but doesn't take into consideration the language structure.
The main advantage of character-based tokenization is that it frees us from these issues related to using subword tokenization on languages with complex morphologies.

Besides the issues related to different morphologies, character-based tokenization is also proposed as a way to avoid choosing between difficult pre-processing tradeoffs, such as: keeping accents, casing, etc. Interestingly, we found that although the CANINE paper claims that *"mBERT initially removed all diacritics, thus eliding tense information in Spanish"*, the official mBERT implementation[3] offers two available pre-trained models: Cased and Uncased. The Uncased model removes any accent markers, whereas in the Cased model the true case and accent markers are preserved. When using mBERT cased, important information encoded in accents (such as diacritics in Spanish) is still kept while using a subword tokenization strategy.

[1] https://huggingface.co/hub
[2] We don't know how to speak Icelandic, the example is taken from https://arxiv.org/abs/2004.07776
[3] https://github.com/google-research/bert/blob/master/multilingual.md

In the next example, we show uncased tokenization is not always appropriate, as we obtain the same token for two different words: “bebe” (he/she drinks), and “bebé” (baby):

“El bebé se bebe la leche” (The baby drinks the milk)

- **Cased tokenization:** ['El', 'be', '##bé', 'se', 'be', '##be', 'la', 'lec', '##he']
- **Uncased tokenization:** ['el', 'bebe', 'se', 'bebe', 'la', 'lech', '##e']

Datasets

For our experiments we use the following datasets. All of them are available in the Hugging Face Hub.

- **glue_mrpc**: Corpus of pairs of sentences in English, annotated to whether the sentences are semantically equivalent or not.
- **ajgt_twitter_ar**: Tweets in Arabic annotated as positive or negative. There are 1350 examples for training and 450 for validation.
- **spanish_diagnostics**: Dataset composed of medical diagnostics, labeled according to whether the data is a dental diagnostic or not. Due to time constraints, we only keep 3500 examples for training and 500 for validation.
- **amazon_reviews_multi**: Amazon product reviews of products written by users. They also include a ranking from 1 to 5. Multilingual dataset, we only considered the Spanish split. The full dataset consists of 200.000 training examples and 5000 for validation and test. We actually trained with the validation dataset and validated with the test dataset for time constraints due to our limitations in hardware. Categories 1 to 5 are balanced, i.e. the same number of examples, 1000, for each one.

Although Arabic and Spanish are not low-resource languages, we found that obtaining good quality datasets for our needs was considerably harder than finding those in English. It is clear that the NLP community is still quite “English-centric”.

Results

In this section we present our numerical results. All the experiments were carried out using pre-trained models loaded from the Hugging Face Hub. We fine-tuned them on the freely available Colab GPUs. In all cases, we trained during 5 epochs with a batch size of 8, Adam optimizer with learning rate 5e-5, and early stopping.

For CANINE, we have two pre-trained models at our disposal. CANINE-S, which was pre-trained with a subword loss, and CANINE-C which was pre-trained with an autoregressive character loss.

CANINE vs mBERT: General performance across different languages

	English		Arabic		Spanish (diagnostics dataset)	
	Accuracy	**F1 Score**	**Accuracy**	**F1 Score**	**Accuracy**	**F1 Score**
canine-s	0.762	0.841	0.762	0.777	0.722	0.712
canine-c	0.809	0.860	**0.822**	**0.817**	**0.896**	**0.879**
mBERT-cased	**0.860**	**0.899**	**0.822**	0.810	0.854	0.831

Table 1: Comparison between mBERT-cased, CANINE-S, and CANINE-C across several datasets in English, Arabic, and Spanish

mBERT-cased vs mBERT-uncased for Spanish text

	spanish_diagnostics		amazon_reviews_multi	
	Accuracy	**F1 Score**	**Accuracy**	**F1 Score**
mBERT-cased	0.854	0.831	0.424	0.443
mBERT-uncased	**0.942**	**0.938**	**0.494**	**0.498**

Table 2: Comparison between mBERT-cased, and BERT-uncased on two datasets in Spanish

In the two Spanish datasets we considered, the uncased version of mBERT got better results than the cased version.

CANINE-Subword-loss vs CANINE-Character-loss

Based on our experiments, CANINE trained with a character-based loss consistently achieved better results than the implementation trained with a subword loss.

Discussion

In general, we found that language-specific datasets for downstream tasks are harder to find for languages other than English. Moreover, the quality of those datasets is lower than their English counterpart. Text examples are less curated and there is a general lack of documentation.

In both Spanish datasets, mBERT-uncased got better results than mBERT-Cased. It seems that having a smaller vocabulary was useful for these tasks, and the information lost due to stripping accents and capital letters was not important for it.

Using CANINE instead of mBERT did not yield a significant improvement in the downstream tasks, even in languages bad-suited for subword tokenization. A possible explanation for this behavior is that "the faults of the tokenizer might be corrected by having enough data". Although mBERT might have worse encodings that destroy part of the linguistic structure of the sentence, given enough data, this information can still be recovered by exploiting statistical information around word contexts. For example, mBERT may be able to distinguish between "bebe" and "bebé" based on the context of their tokens without explicitly taking into account the accent in "é" during tokenization. Nevertheless, we acknowledge that our experiments are not exhaustive and they were restricted to what we could train on Colab GPUs. After completing all the experiments we found that in the CANINE conference paper[4], specifically the results for Arabic and Spanish obtained by the authors were worse with CANINE than with mBERT.

However, we do believe that having a tokenizer-free and vocabulary-free model is advantageous since it eliminates the need for a hand-engineered tokenization procedure. This allows the final model to be more flexible to different input languages and remove arbitrary pre-preprocessing of text data, while still getting comparable performance to other subword SOTA models.

Statement of contribution

Nicolás coded the training setup from which we based all further experiments. David conducted most of the experiments regarding CANINE. Elías did research related to languages affected by subword tokenization. The rest of the workload was equally distributed.

[4]https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00448/109284/Canine-Pre-training-an-Efficient-Tokenization-Free