Baltic J. Modern Computing, Vol. 11 (2023), No. 4, pp. 591–606
https://doi.org/10.22364/bjmc.2023.11.4.05

On Checking Robustness on Named Entity Recognition with Pre-trained Transformers Models

Aitor GARCÍA-PABLOS[1], Justina MANDRAVICKAITĖ[2], Egidija VERŠINSKIENĖ[2]

[1] Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009 Donostia-San Sebastián, Spain
[2] Lithuanian Cybercrime Center of Excellence for Training, Research & Education (L3CE), Didlaukio g. 55, Vilnius, Lithuania

agarciap@vicomtech.org, justina@l3ce.eu, egidija@l3ce.eu

ORCID 0000-0001-9882-7521, ORCID 0000-0001-9426-6165, ORCID 0000-0002-1303-9567

Abstract. In this paper we are conducting a series of experiments with several state-of-the-art models, based on Transformers architecture, to perform Named Entity Recognition and Classification (NERC) on text of different styles (social networks vs. news) and languages, and with different levels of noise. We are using different publicly-available datasets such as WNUT17, CoNLL2002 and CoNLL2003. Furthermore, we synthetically add extra levels of noise (random capitalization, random character additions/replacements/removals, etc.), to study the impact and the robustness of the models. The Transformer models we compare (mBERT, CANINE, mDeBERTa) use different tokenisation strategies (token-based vs. character-based) which may exhibit different levels of robustness towards certain types of noise. The experiments show that the subword-based models (mBERT and mDeBERTa) tend to achieve higher scores, especially in the presence of clean text. However, when the amount of noise increases, the character-based tokenisation exhibits a smaller performance drop, suggesting that models such as CANINE might be a better candidate to deal with noisy text.

Keywords: NER, Transformers, model robustness, English, Dutch, Spanish

1 Introduction

Natural Language Processing (NLP) models are essential in processing and understanding vast amounts of textual data. Transformers, a type of NLP model, has recently shown very good results in various NLP tasks. However, the performance of these models can be impacted by noise in the input text, such as typos and grammatical errors, which are common in social media and internet-based text.

Transformers are State-of-the-Art Deep Learning models that can be pre-trained on large amounts of unlabelled data and later fine-tuned for particular tasks. As pre-training phase usually involves a large amount of data and computation power, it can usually be afforded by large companies, e.g., Google, Microsoft, or Facebook. Thus the number of pre-trained models available for experimentation is limited. When it comes to multilingual models, the list of available models gets reduced further. Apart from the source of pre-training data, the models differ in other aspects, such as the text-tokenisation strategy. Therefore, getting insight and understanding the performance of some of these modern models in the presence of noisy text would guide the choice of each model for the proper task.

In this work we compare three well-known Transformer-based models (CANINE, mBERT and mDeBERTA) on four different datasets which span three different languages (English, Spanish, and Dutch). We added controlled synthetic noise to these datasets, obtaining three different variations with incremental noise of different types (capitalisation noise and character-editing noise). We ran experiments with these three models on all of the resulting datasets and made a comparison of the resulting scores and the degradation of the scores with regard to the added noise. The results show that the subword-based models tend to work better on formal and clean text, but when the amount of noise increases, the character-based tokenisation exhibits a more robust behaviour, reducing the amount of degradation in the resulting scores.

The rest of this paper is structured as follows. Section 2 briefly describes the related work. Section 3 details the datasets we have used for the evaluation and comparison. Section 4 illustrates the noise addition process we have followed to obtain different datasets with a controlled incremental level of text noise. Section 5 introduces the pre-trained Transformers models that we have included in the evaluation and comparison. Section 6 describes the experimental setting and the training details from which we have obtained the results. Section 7 contains the evaluation, including the tables with quantitative results and their description. Finally, 8 contains the concluding remarks and future work.

2 Related Work

State-of-the-art NER models usually work well on datasets written in a standard language with accurate grammar and similar to the data they have been trained on. Their performance drops on noisy datasets, particularly those with capitalization problems (Mayhew et al., 2020; Bodapati et al., 2019). Also, traditional NER models make predictions based on the features of the tokens and the context. As confusion among different entity categories exists, NER models could overfit to the source domain entities and generalise worse to the target domain (Liu et al., 2020). It has been shown that even Transformer-based models, such as BERT, perform worse in the case of synonym swaps or spelling mistakes in a sentence (Jin et al., 2020; Hsieh et al., 2019; Sun, Hashimoto, Yin, Asai, Li, Yu and Xiong, 2020).

Self-augmentation appeared to be a perspective solution to this problem (Zhang et al., 2018; Wei and Zou, 2019; Dai and Adel, 2020; Chen, Wang, Tian, Yang and Yang, 2020; Karimi et al., 2021). It includes automatic generation of a pseudo-training

dataset derived from the original training data with golden labels (Wu et al., 2022). Efforts to decrease dependency on labeled data for NLP tasks include token-level adversarial attacks (e.g., word substitutions (Kobayashi, 2018; Wei and Zou, 2019; Dai and Adel, 2020; Zeng et al., 2020) or addition of noise (Narayan et al., 2019)) and paraphrasing at sentence-levels (e.g., back translations (Xie et al., 2020; Sennrich et al., 2016a; Fadaee et al., 2017; Dong et al., 2017; Yu et al., 2018)). The latter includes sentence-level rewriting without significantly altering semantics. Another noteworthy technique is mixup, which uses the feature-level data augmentation (Zhang et al., 2018; Chen, Wang, Tian, Yang and Yang, 2020; Sun, Xia, Yin, Liang, Philip and He, 2020; Zhang et al., 2020). It was originally proposed in computer vision and used for implementing linear interpolations between randomly sampled image pairs to create virtual training data. Later, the idea was adapted to the textual domain (Miao et al., 2020; Chen, Yang and Yang, 2020) and applied to text classification (Chen, Wang, Tian, Yang and Yang, 2020).

One more proposed data augmentation technique is the constrained augmentation such as contextual augmentation (Kobayashi, 2018), conditional BERT (Wu et al., 2019) and AUG-BERT (Shi et al., 2019). For a task, for which a model is taught on labeled data, constrained augmentation transforms a pre-trained language model into a label-conditional language model (Bari et al., 2021). Also, it was proposed to create entity-switched datasets by replacing entities with others of the same type (Agarwal et al., 2020). Meanwhile, augmenting training data with upper- and lower-cased text variations was suggested in (Bodapati et al., 2019; Vīksna and Skadiņa, 2021a) to reduce the influence of noisy data on NER performance.

Linguistic noise such as word dropout and synonym replacement performs as well as statistical noise while being simpler and easier to fine-tune (Narayan et al., 2019). In (Viksna and Skadiņa, 2021b) three different strategies to increase the robustness of NER have been explored, namely, error injection into grammatically correct texts, augmenting grammatically correct texts with faulty texts, and augmenting grammatically correct texts with faulty texts with specific errors.

In this work we choose to evaluate three different Transformers-based models. The Transformers architecture is a deep neural network architecture that is used in most of the current State-of-the-Art Deep Learning models. The success of this type of architecture is due to the self-attention mechanism that lies at the core of the architecture which allows the models to attend and operate on the whole input sequence at once, thus enabling the models to learn long-term dependencies (Zuo et al., 2020), syntactic and semantic relations (Sajjad et al., 2022), etc. On the other hand, transformers can be pre-trained on large amounts of raw text, i.e., use self-supervised training, which allows them to learn language models from scratch, without labeled data (Chen et al., 2021).

The choice of models for comparison also takes into consideration the way the model vocabulary is generated. The vocabulary will contain the minimum units the model will manipulate when modelling the input text (Wies et al., 2021). Depending on the algorithm used to generate the vocabulary, the way the model splits the text into tokens (words, sub-words, punctuation marks, etc.) will vary. Thus tokenisation is important for language models to segment a raw text string into a sequence for the

input (Devlin et al., 2019; Brown et al., 2020; Wolf et al., 2020). For this step, several strategies can be applied.

Classic NLP tools used full words as tokens. The advantage of this strategy is having one exact representation for each word in the vocabulary, which is easier to process. However, it has several drawbacks, e.g., the need for a very large vocabulary to cover representative amount of the words from a given language, the over-specialisation of a vocabulary (i.e. the model cannot be transferred from one language to another because most of the tokens/words will not provide meaningful information) (Wieneke et al., 2020), and data sparsity (i.e., many words will not have enough occurrences in the training data).

The Transformer-based models use more modern tokenisation strategies that split the words into subwords, using different algorithms to derive the most appropriate subword-vocabulary space in a given corpus, such as WordPiece, Byte-Pair Encoding (BPE) or SentencePiece (Rai and Borah, 2021; Sennrich et al., 2016b; Kudo and Richardson, 2018). Another option is to tokenise at the character level so that each input character is treated as a token that is represented in the model. The character-based approach gives more flexibility, although it is more difficult for the model to understand the meaning of a word from its individual character tokens (Lees et al., 2022).

In this work we compare three different Transformer-based models that are widely used in the community: CANINE, mBERT and mDeBERTaV3. Each one uses a different tokenisation strategy: WordPiece, SentencePiece and character-based tokenisation. All three of them are multilingual. We train, evaluate and compare these three models for a NERC task on four different datasets covering three different languages and three different levels of extra synthetic noise. The objective is to examine the robustness of each model and to conclude under which circumstances each model and tokenisation strategy may perform better.

3 Datasets

For the evaluation and comparison of the three models, we use several NER datasets. NER consists of detecting and classifying words in a text that are the mentions of a certain entity, e.g., locations, people names, or organisations. The models need to accurately tell which token belongs to each possible entity type or category, and perturbations in the input data may complicate the ability of the model to perform the task. The datasets used in our experimentation are shortly described in the following subsections.

3.1 WNUT'17 Dataset

We used WNUT'17 Shared Task[3] dataset as one of the datasets to study the robustness of pre-trained models in NER task. The focus of this shared task – unusual and rare entities in noisy text (Derczynski et al., 2017). WNUT'17 dataset is annotated and made of 2,295 texts taken from different sources – Reddit, Twitter, YouTube, and StackExchange comments. This dataset is split into training, development and test subsets.

[3] For more information, check `https://noisy-text.github.io/2017/index.html`

Training data consists of 1000 annotated tweets. The development subset is made of Youtube data (user-generated comments) and the test subset includes data from Reddit and StackExchange. WNUT'17 dataset is annotated with 6 classes:

1. Person
2. Location (includes GPE and facilities)
3. Corporation
4. Product (goods and services)
5. Creative work (e.g., song, movie, book, and so on)
6. Group (music bands, sports teams, and non-corporate organisations)

Table 1. Statistics of final development and test subsets Derczynski et al. (2017).

	Train	**Dev**	**Test**
Documents	1000	1008	1287
Tokens	65124	15,734	23,394
Entities	1984	835	1040

As a preprocessing step, common entities were filtered out from development and test subsets. WNUT'17 dataset was constructed in such a way so it would provide high-variance data i.e., have very few repeated surface forms. The statistics of final development and test subsets are presented in Table 1.

In the WNUT'17 shared task the score achieved by the best-performing participant (Jansson and Liu, 2017) was 39.98% of F-score for NERC and 37.77% of F-score for the Surface Forms metric (which takes into account only the set of different entities detected/omitted regardless of their frequency).

3.2 CoNLL 2002 & 2003 Datasets

CoNLL-2002 Shared Task focused on language-independent NER. The annotated types of named entities include persons, locations, organizations and miscellaneous (entities that do not belong to the other 3 classes). The data cover two languages – Dutch and Spanish. The Spanish data is a collection of news wire articles made available by the *Spanish EFE News Agency* (the articles are from May 2000) while Dutch data consist of four editions of the Belgian newspaper *De Morgen* of 2000 (June 2, July 1, August 1 and September 1) Tjong Kim Sang (2002). Each of the languages has a training subset, a development subset and a test subset. Table 2 and 3 show the statistics of Spanish and Dutch datasets respectively.

In the original CoNLL-2002 shared-task, the scores achieved by the best-performing participant (Carreras et al., 2002) were 81.39% of F-score for Spanish and 77.05% of F-score for Dutch.

CoNLL-2003 Shared Task concentrated on language-independent NER as well. It covered English and German languages. For each language there were annotated subsets (training, development and test subset) and a large subset with unannotated data

Table 2. Statistics of CoNLL-2002 Shared Task Spanish data.

Spanish data	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	8323	264715	4913	2173	7390	4321
Development set	1915	52923	984	445	1700	1222
Test set	1517	51533	1084	339	1400	735

Table 3. Statistics of CoNLL-2002 Shared Task Dutch data.

Dutch data	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	15806	202931	3208	3338	2082	4716
Development set	2895	37761	479	748	686	703
Test set	5195	68994	774	1187	882	1098

(Tjong Kim Sang and De Meulder, 2003). The English data was taken from the Reuters Corpus and consists of Reuters news stories between August 1996 and August 1997. Table 4 shows the English dataset distribution.

In the original CoNLL-2003 shared task the score achieved by the best-performing participant (Florian et al., 2003) was 88.76% of F-score for English. It's important to note that these results are no longer considered State-of-the-Art, as these datasets have been extensively utilized for research across numerous papers, achieving progressively higher scores. The current leading score is expected to surpass 90%, depending on the configuration and language specifics. For instance, in the original BERT paper (Devlin et al., 2019), a 92.6% F-score is reported for the English CoNLL dataset using a sole English BERT base model.

In our experiments we decided to use only the datasets for three languages (English, Spanish and Dutch).

4 Noise addition methods

For all the described datasets we have conducted a series of synthetic noise additions to systematically perturb the original words making the text understanding task more challenging for the models. The result of these additions is a set of derived datasets with an increasing amount of noise of different types.

We base our synthetic noise additions on the works of (Bodapati et al., 2019; Rychalska et al., 2019; Náplava et al., 2021; Vīksna and Skadiņa, 2021a; Viksna and Skadiņa, 2021b). We chose a mixture of perturbation methods according to the set of languages the datasets are written in as the same set was applied to all of them. Also, we took into consideration the potential the noise additions have in order to address the most common errors and irregularities which may be the cause of lesser robustness in NER models, especially when they are applied to informal texts. We chose to add linguistic and statistical noise to the datasets in our experiments because this type of noise injection is easier to fine-tune (Narayan et al., 2019).

Table 4. Statistics of CoNLL-2003 Shared Task English data.

English data	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	946	14,987	203,621	7140	3438	6321	6600
Development set	216	3,466	51,362	1837	922	1341	1842
Test set	231	3,684	46,435	1668	702	1661	1617

Informal texts on the Internet, such as the ones from social media and chats, usually contain common writing errors related to a lack of proper capitalisation, orthographic errors and typing errors. To represent this kind of noise, we have included synthetic noise of two types: capitalisation-related noise and word-edit noise. The former alters the capitalisation of the words, setting them to lowercase or uppercase. This kind of added noise tries to mimic the fact that the user on the Internet, when writing informal texts, pays less attention to proper capitalisation or omits it totally.

The other noise addition alters individual words by adding, removing, switching or replacing single characters. This perturbation tries to emulate typos, misspellings, or shortening of words and slang. To adequately emulate some of these phenomena a more sophisticated common error study would be necessary, but we assume that this simpler approach should suffice to measure the robustness of each model against input text degradation.

The configuration of noise addition is as follows:

- Capitalisation noise: for any given word, with a probability of 10%, set the word to lowercase, set the word to uppercase, or randomly capitalize one of its characters.
- Word-edit noise: for any given word, with a probability of 5%, duplicate a character, or remove a character, or replace the character, or swap a character with the following one.

For each original dataset we generated three perturbed variants. One with just capitalisation noise, another with just word-edit noise, and a third one combining both noise-addition strategies. The size and label distribution of the resulting datasets are identical to the original ones, since the perturbations apply to individual words, keeping the number of sentences and labeled entities the same.

5 Pre-trained Models

We compare three different publicly available Transformers-based models that are well-known by the community: CANINE, mBERT and mDeBERTaV3. Each one uses a different tokenisation strategy: character-based, WordPiece and SentencePiece tokenisation respectively. All three models are multilingual, allowing us to evaluate their robustness across different languages. Also, it is worth noting that the three models we used are of an equivalent and/or comparable size: a "base" model of 12 stacked Transformer layers, as opposed to larger models such as BERT-large and others. Furthermore, the three models are "cased" models, which means that the capitalisation is relevant to them, as opposed to "uncased" models that work only with lowercase text. These three models, selected for our study, are briefly introduced in the following subsections.

5.1 CANINE

CANINE is a character-based neural encoder (Clark et al., 2022). Its major difference from other Transformer-based text encoders is that it is trained directly at a Unicode character-level. Training at a character-level comes with a longer sequence length, which CANINE solves with a downsampling strategy, before applying a deep Transformer encoder. Model input is sequences of Unicode characters, that give a much larger flexibility and adaptability to other languages.

CANINE has the following main components (Clark et al., 2022):

1. a vocabulary-free technique for embedding text;
2. a character-level model which makes CANINE by downsampling and upsampling (changing the length of input strings);
3. a masked language modeling on a character-level model.

The CANINE pre-training procedure uses masked-language modelling (MLM) combined with one of two loss calculation techniques – autoregressive character prediction (Yang et al., 2019) or subword prediction (Joshi et al., 2020) – for tokenization-free model which follows the pre-training.

While recent tokenizers, which are based on a data-derived subword approach, are more robust compared to rule-based tokenizers, these techniques still face limitations due to the unique characteristics of different languages (Clark et al., 2022). Furthermore, models with fixed vocabularies are constrained in their ability to adapt (Xue et al., 2022). CANINE addresses these issues by adopting a tokenization-free and vocabulary-free approach. In the case of CANINE, tokens for a given example text, such as 'Jim Henson was a famous puppeteer,' would be represented as:

```
['J', 'i', 'm', ' ', 'H', 'e', 'n', 's', 'o', 'n', ' ', 'w',
    'a', 's', ' ', 'a', ' ', 'f', 'a', 'm', 'o', 'u', 's', '
    ', 'p', 'u', 'p', 'p', 'e', 't', 'e', 'e', 'r']
```

Note how even the individual white spaces are represented as part of the input.

5.2 mBERT

The most widely known Transformer encoder is the original BERT model (Devlin et al., 2019) published by Google, which helped to popularise the Transformer architecture. Particularly, its multilingual version, known as mBERT, enabled the community to experiment and contribute to the State-of-the-Art in many tasks and for different languages. Multilingual BERT (mBERT) has been trained on 104 languages and shown a good performance on several NLP tasks in cross-lingual settings (Pires et al., 2019; Wu and Dredze, 2020). It can be seen as the grandfather of the Transformer-based models, but it is still a referent to measure against.

With regard to tokenisation, mBERT uses WordPiece tokenisation strategy, which calculates a subword vocabulary based on provided training data. Using mBER tokeniser the following example sentence "Jim Henson was a famous puppeteer" results in the tokens:

```
['Jim', 'Hen', '##son', 'is', 'a', 'famous', 'pu', '##ppet',
    '##eer']
```

Some words are split into subwords, and the subwords that are not part of the beginning of a word are marked with '##'. White spaces and line breaks are not represented. Usually, WordPiece algorithm is no longer used by new models, favoring the use of SentencePiece tokenisation.

5.3 mDeBERTa

mDeBERTa is a DeBERTaV3 (He et al., 2021) language model pre-trained by Microsoft on multilingual *Common Crawl* data. Instead of masked language modeling (MLM) it used replaced token detection (RTD) which is more efficient for pre-training (He et al., 2021). Thus mDeBERTa combines DeBERTa (He et al., 2020) with ELECTRA (Clark et al., 2020) and results in increased performance. The tokenisation used by mDeBERTa, similar to other modern Transformer-based text-models, is SentencePiece. Applied to the example text "Jim Henson was a famous puppeteer" the mDeBERTa would produce:

```
['_Jim', '_Hen', 'son', '_is', '_', 'a', '_', 'famous', '_pup
    ', 'pete', 'er']
```

The head subwords that start right after a white space are marked with a special symbol, while the subsequent subwords are not.

6 Experimental setup

For the robustness study of selected models, we used 4 experiments with different configurations, where we added different levels of noise to the data. More precisely, we have had the following experimental setups:

1. **noise0:** the original dataset unperturbed
2. **noise1:** added random capitalization noise to the dataset
3. **noise2:** added random character addition, removal, duplicating, switching, etc.
4. **noise3:** noise1 + noise2 combined

For each dataset and its noisy variants, we have trained each model using the same hyper-parameters. Table 5 describes the hyper-parameters related to the training of the models.

Each training uses the official training, validation and test split (with the corresponding amount of noise in each case). The gold-labels for each dataset remain exactly the same regardless of the noise addition. The training is configured to run a maximum number of epochs using an early-stopping strategy with specified patience. If the validation score of the model does not improve for the specified number of epochs, the training stops.

For each training run, we select the model checkpoint that performs the best on its corresponding validation set. Then, we evaluate each model on the corresponding test

Table 5. Training Hyper-parameters Shared by All Experiments

Hyper-parameter	Value
Max training epochs	100
Early stopping patience	20
Batch size	4
Learning rate	2E-5
Warm-up epochs	4
Gradients accumulation steps	4

set to calculate various quantitative metrics and compare the results. It is worth noting that, in addition to comparing the models against each other, we were also interested in evaluating their robustness to noise. This means that we wanted to compare each model to itself when trained on different types of noisy data. The results of the comparison of the models are presented in the following section.

7 Evaluation

In order to evaluate and compare the models, each experiment has been run 3 times with different random seeds (42, 54 and 86), to average the results and obtain a more robust conclusion (since variations on randomized initialization on the models may result in variations in the final performance).

Table 6 summarizes the evaluation of three models for each noise-variant version of each dataset. The table shows two different metrics, a classical F1-score, and a surface-forms metric. The surface forms metric takes into account only the set of surface forms detected or missed, without taking into account their frequency in the dataset, reducing the bias from detecting very frequent and easy entities versus scarce but more challenging entities. This metric was described and used in the original WNUT17 shared task (Derczynski et al., 2017).

Table 7 shows the average standard deviation of all the experiments for each model, which were calculated from the 3 runs per experiment for each model and dataset variation combination. The values are below 0.5% of the metrics' scores, which indicates that the measured scores are stable and not biased by a lucky or unlucky model initialisation.

For the WNUT17 dataset we observe that the F-score is around 40%. It is something to be expected for this dataset since even the original version (without synthetic noise) is noisy and challenging. In the original WNUT17 shared task (Derczynski et al., 2017) the best-performing system obtains 41% of F-score being a carefully crafted system, while Transformers-based models obtain a similar or even higher score without any specific feature engineering. The mDeBERTa model shows much higher scores than the other two in the original dataset, with a 48.3% of F-score and a 46.8% of surface-forms metric. In the WNUT17 evaluation we see that all three models experience a similar drop in performance when the extra noise is added though it needs to be taken into consideration the original version of the WNUT17 dataset is already noisy.

We observe that for CANINE, the noise related to character edition (noise2) hurts its performance more than the capitalisation-related noise. It makes sense since CANINE treats each character individually, and it can understand the relation between a capital letter and its lowercase counterpart, while for the subword-based models such a change in the input may lead to a totally different tokens sequence.

For the CoNLL datasets we observe a different scenario in comparison to the WNUT17 dataset. All the models obtain very high results, as expected. If we observe the degradation of the performance with the noise addition, we observe that CANINE exhibits a more robust behaviour. From the score obtained in the original, unperturbed, CoNLL English dataset (90.1% F1-score) to the score obtained in the noisiest version (87.8%) there is a drop of 2.3 points. For mBERT and mDeBERTa the drop is 5.2 and 5 points respectively. The surface-forms score shows similar behaviour.

For the CoNLL Spanish the results are equivalent among the models, showing that CANINE is not the best performing model when the data is clean but is the best resisting the extra noise addition. For the CoNLL Dutch dataset the result is not that clear, being that the performance degradation due to the noise addition is similar for all the models, but mDeBERTa obtained better overall scores.

Table 6. Evaluation results of NER models on various dataset variations and datasets, showing the F-score scores of the NERC and Surface Forms for the original datasets and the score variation (i.e. degradation) for each noise-added variant.

Model	**Noise**	**WNUT17**		**CoNLL English**		**CoNLL Spanish**		**CoNLL Dutch**	
		NERC	**Surf.**	**NERC**	**Surf.**	**NERC**	**Surf.**	**NERC**	**Surf.**
CANINE	original	41.3	40.8	90.1	88.5	85.7	84.3	87.8	84.7
	noise1	-4.2	-3.8	-0.6	0.1	-1.3	-1.1	-3.1	-2.2
	noise2	-5.5	-5.6	-1.1	-0.9	-0.8	-1.2	-2.1	-1.9
	noise3	-8.1	-7.5	-2.3	-1.8	-2.2	-2.5	-6.0	-5.3
mBERT	original	44.1	43.1	90.7	88.5	87.1	86.0	90.4	87.8
	noise1	-6.0	-5.6	-2.9	-2.2	-4.5	-5.4	-6.4	-6.9
	noise2	-4.6	-4.8	-2.0	-1.9	-1.6	-2.6	-3.3	-3.8
	noise3	-8.9	-8.4	-5.2	-4.3	-7.0	-7.8	-9.1	-9.5
mDeBERTa	original	48.3	46.8	88.3	85.3	88.2	86.7	92.2	89.9
	noise1	-4.1	-3.9	-2.3	-1.9	-1.8	-1.8	-3.4	-3.2
	noise2	-3.9	-4.0	-1.8	-1.3	-1.4	-1.7	-2.5	-2.8
	noise3	-6.7	-6.6	-5.0	-4.6	-3.6	-3.7	-5.6	-5.7

According to these results, there is no best model for all situations. The initial knowledge of the language model (which depends on its pre-training data and strategy) and the type of tokenisation favour different scenarios. The subword-based models, which are the most widely used in the community nowadays, have the advantage of having more informative tokens when the content matches the tokenisation they have

Table 7. Average standard deviation for NERC and Surface Forms scores across models.

Model	NERC	Surface Forms
CANINE	0.41	0.46
mBERT	0.39	0.45
mDeBERTa	0.46	0.39

been trained with. Modelling a text character by character not only is more computationally expensive but also increases the difficulty of deriving the meaning of a word and its contexts by composing a very individual character, that bears no meaning on its own. However, character-based tokenisation provides more flexibility and robustness against certain perturbations in the input. The altering of an individual character or the capitalisation of a single letter only affects that very character instead of potentially disrupting the whole token sequence.

8 Discussion and Conclusions

In this paper we have compared three different State-of-the-art Transformers models: CANINE, mBERT and mDeBERTA. The selection of these three models is based on the fact that all of them are multilingual language models, well-known in the community, and each of them uses a different tokenisation algorithm or paradigm. In particular CANINE is a character-based model that does not require any specific tokenisation. This kind of models is very interesting due to their flexibility for modelling any kind of input, but they are not as extended as their subword-based counterparts, probably due to the extra computational cost of treating a sequence character by character.

For the comparison in terms of robustness against noise, we have taken four different publicly available NERC datasets, spanning three different languages. We have created noisy versions of each of them. In particular, we have generated different levels of noise of different types: capitalisation and character-editing noise. The result is a set of four original datasets plus three noisy variants of each of them.

We have trained the compared models on each of the resulting datasets, using the exact same procedure and hyper-parameters, to compare the results side by side and extract conclusions. The main conclusion is that, although there is no single best model, the character-based nature of CANINE seems to be more robust against input corruption. All the models' experiment drops in their performances, but CANINE shows smaller drops compared to the other models.

The State-of-the-Art in NLP has been evolving really fast over the last couple of years, and every year brings a new set of models. Therefore in the future we plan to extend this comparison to other types of NLP tasks. We also prepare to explore the differences of encoder-decoder models for text-generation tasks, such as the ByT5, based on characters, versus other models from the T5 family based on subword tokenisation. It would be useful to assess to which extent the character-based tokenisation makes more robust the text-generation tasks in the presence of noise, or when the output needs an extra layer of flexibility. Furthermore, it would be interesting to conduct a similar study

for languages such as Estonian, Latvian, or Lithuanian, and observe how the models perform.

Finally, as of the time of writing, the landscape of NLP has undergone a significant transformation with the emergence of truly Large Language Models (LLMs) containing billions of parameters. The computational demands associated with training and deploying such models can be prohibitive or counterproductive, highlighting the ongoing relevance of smaller, more specialized models like those explored in this study, particularly for specific tasks and use cases. In light of this, exploring the resilience of LLMs in the presence of noisy input could prove to be a valuable avenue for future research.

Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and innovation programme under project GRACE, grant agreement No. 883341.

References

Agarwal, O., Yang, Y., Wallace, B. C., Nenkova, A. (2020). Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models, *arXiv preprint arXiv:2004.04123* .

Bari, M. S., Mohiuddin, M. T., Joty, S. (2021). Uxla: A robust unsupervised data augmentation framework for zero-resource cross-lingual nlp, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1978–1992.

Bodapati, S., Yun, H., Al-Onaizan, Y. (2019). Robustness to capitalization errors in named entity recognition, *arXiv preprint arXiv:1911.05241* .

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. (2020). Language models are few-shot learners, *Advances in neural information processing systems* **33**, 1877–1901.

Carreras, X., Màrques, L., Padró, L. (2002). Named entity extraction using adaboost, *Proceedings of CoNLL-2002*, Taipei, Taiwan, pp. 167–170.

Chen, J., Wang, Z., Tian, R., Yang, Z., Yang, D. (2020). Local additivity based data augmentation for semi-supervised ner, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1241–1251.

Chen, J., Yang, Z., Yang, D. (2020). Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2147–2157.

Chen, S., Huang, S., Pandey, S., Li, B., Gao, G. R., Zheng, L., Ding, C., Liu, H. (2021). Et: re-thinking self-attention for transformer models on gpus, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–18.

Clark, J. H., Garrette, D., Turc, I., Wieting, J. (2022). Canine: Pre-training an efficient tokenization-free encoder for language representation, *Transactions of the Association for Computational Linguistics* **10**, 73–91.

Clark, K., Luong, M.-T., Le, Q. V., Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators, *arXiv preprint arXiv:2003.10555* .

Dai, X., Adel, H. (2020). An analysis of simple data augmentation for named entity recognition, *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3861–3867.

Derczynski, L., Nichols, E., van Erp, M., Limsopatham, N. (2017). Results of the wnut2017 shared task on novel and emerging entity recognition, *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 140–147.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of NAACL-HLT*, pp. 4171–4186.

Dong, L., Mallinson, J., Reddy, S., Lapata, M. (2017). Learning to paraphrase for question answering, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 875–886.

Fadaee, M., Bisazza, A., Monz, C. (2017). Data augmentation for low-resource neural machine translation, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 567–573.

Florian, R., Ittycheriah, A., Jing, H., Zhang, T. (2003). Named entity recognition through classifier combination, *in* Daelemans, W., Osborne, M. (eds), *Proceedings of CoNLL-2003*, Edmonton, Canada, pp. 168–171.

He, P., Gao, J., Chen, W. (2021). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, *arXiv preprint arXiv:2111.09543* .

He, P., Liu, X., Gao, J., Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention, *arXiv preprint arXiv:2006.03654* .

Hsieh, Y.-L., Cheng, M., Juan, D.-C., Wei, W., Hsu, W.-L., Hsieh, C.-J. (2019). On the robustness of self-attentive models, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1520–1529.

Jansson, P., Liu, S. (2017). Distributed representation, LDA topic modelling and deep learning for emerging named entity recognition from social media, *Proceedings of the 3rd Workshop on Noisy User-generated Text*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 154–159.
https://aclanthology.org/W17-4420

Jin, D., Jin, Z., Zhou, J. T., Szolovits, P. (2020). Is bert really robust? a strong baseline for natural language attack on text classification and entailment, *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34, pp. 8018–8025.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans, *Transactions of the Association for Computational Linguistics* **8**, 64–77.

Karimi, A., Rossi, L., Prati, A. (2021). Aeda: An easier data augmentation technique for text classification, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2748–2754.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations, *Proceedings of NAACL-HLT*, pp. 452–457.

Kudo, T., Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71.

Lees, A., Tran, V. Q., Tay, Y., Sorensen, J., Gupta, J., Metzler, D., Vasserman, L. (2022). A new generation of perspective api: Efficient multilingual character-level transformers, *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3197–3207.

Liu, Z., Winata, G. I., Fung, P. (2020). Zero-resource cross-domain named entity recognition, *arXiv preprint arXiv:2002.05923* .

Mayhew, S., Nitish, G., Roth, D. (2020). Robust named entity recognition with truecasing pretraining, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 8480–8487.

Miao, Z., Li, Y., Wang, X., Tan, W.-C. (2020). Snippext: Semi-supervised opinion mining with augmented data, *Proceedings of The Web Conference 2020*, pp. 617–628.

Náplava, J., Popel, M., Straka, M., Straková, J. (2021). Understanding model robustness to user-generated noisy texts, *arXiv preprint arXiv:2110.07428* .

Narayan, P. L., Nagesh, A., Surdeanu, M. (2019). Exploration of noise strategies in semi-supervised named entity classification, *8th Joint Conference on Lexical and Computational Semantics,* SEM@ NAACL-HLT 2019*, Association for Computational Linguistics (ACL), pp. 186–191.

Pires, T., Schlinger, E., Garrette, D. (2019). How multilingual is multilingual bert?, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4996–5001.

Rai, A., Borah, S. (2021). Study of various methods for tokenization, *Applications of Internet of Things*, Springer, pp. 193–200.

Rychalska, B., Basaj, D., Gosiewska, A., Biecek, P. (2019). Models in the wild: On corruption robustness of neural nlp systems, *International Conference on Neural Information Processing*, Springer, pp. 235–247.

Sajjad, H., Durrani, N., Dalvi, F., Alam, F., Khan, A., Xu, J. (2022). Analyzing encoded concepts in transformer language models, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3082–3101.

Sennrich, R., Haddow, B., Birch, A. (2016a). Improving neural machine translation models with monolingual data, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 86–96.

Sennrich, R., Haddow, B., Birch, A. (2016b). Neural machine translation of rare words with subword units, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Berlin, Germany, pp. 1715–1725.
https://aclanthology.org/P16-1162

Shi, L., Liu, D., Liu, G., Meng, K. (2019). Aug-bert: An efficient data augmentation algorithm for text classification, *International Conference in Communications, Signal Processing, and Systems*, Springer, pp. 2191–2198.

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

Sun, L., Xia, C., Yin, W., Liang, T., Philip, S. Y., He, L. (2020). Mixup-transformer: Dynamic data augmentation for nlp tasks, *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3436–3440.

Tjong Kim Sang, E. F. (2002). Introduction to the conll-2002 shared task: Language-independent named entity recognition, *Proceedings of CoNLL-2002*, Taipei, Taiwan, pp. 155–158.

Tjong Kim Sang, E. F., De Meulder, F. (2003). Introduction to the conll-2003 shared task: language-independent named entity recognition, *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pp. 142–147.

Vīksna, R., Skadiņa, I. (2021a). Multilingual slavic named entity recognition, *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pp. 93–97.

Viksna, R., Skadiņa, I. (2021b). Robustness of named entity recognition: Case of latvian, *International Conference on Statistical Language and Speech Processing*, Springer, pp. 50–58.

Wei, J., Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks, *Proceedings of the 2019 Conference on Empirical Methods in Nat-*

ural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6382–6388.

Wieneke, L., Kalyakin, R., Biryukov, M., Andersen, E. (2020). How to read the 52.000 pages of the british journal of psychiatry? a collaborative approach to source exploration, *Journal of Data Mining & Digital Humanities* .

Wies, N., Levine, Y., Jannai, D., Shashua, A. (2021). Which transformer architecture fits my data? a vocabulary bottleneck in self-attention, *International Conference on Machine Learning*, PMLR, pp. 11170–11181.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. et al. (2020). Transformers: State-of-the-art natural language processing, *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45.

Wu, L., Xie, P., Zhou, J., Zhang, M., Ma, C., Xu, G., Zhang, M. (2022). Robust self-augmentation for named entity recognition with meta reweighting, *CoRR* .

Wu, S., Dredze, M. (2020). Are all languages created equal in multilingual bert?, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 120–130.

Wu, X., Lv, S., Zang, L., Han, J., Hu, S. (2019). Conditional bert contextual augmentation, *International conference on computational science*, Springer, pp. 84–95.

Xie, Q., Dai, Z., Hovy, E., Luong, T., Le, Q. (2020). Unsupervised data augmentation for consistency training, *Advances in Neural Information Processing Systems* **33**, 6256–6268.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., Raffel, C. (2022). Byt5: Towards a token-free future with pre-trained byte-to-byte models, *Transactions of the Association for Computational Linguistics* **10**, 291–306.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding, *Advances in neural information processing systems* **32**.

Yu, A. W., Dohan, D., Le, Q., Luong, T., Zhao, R., Chen, K. (2018). Fast and accurate reading comprehension by combining self-attention and convolution, *International Conference on Learning Representations*, Vol. 2.

Zeng, X., Li, Y., Zhai, Y., Zhang, Y. (2020). Counterfactual generator: A weakly-supervised method for named entity recognition, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7270–7280.

Zhang, H., Cisse, M., Dauphin, Y. N., Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization, *International Conference on Learning Representations*.

Zhang, R., Yu, Y., Zhang, C. (2020). Seqmix: Augmenting active sequence labeling via sequence mixup, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8566–8579.

Zuo, S., Jiang, H., Li, Z., Zhao, T., Zha, H. (2020). Transformer hawkes process, *International conference on machine learning*, PMLR, pp. 11692–11702.

Received May 3, 2023 , revised August 28, 2023, accepted October 12, 2023