CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters

Hicham El Boukkouri[1], Olivier Ferret[2], Thomas Lavergne[1], Hiroshi Noji[3], Pierre Zweigenbaum[1], Junichi Tsujii[3]

[1]Université Paris-Saclay, CNRS, LIMSI, Orsay, France,
[2]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France,
[3]Artificial Intelligence Research Center (AIRC), AIST, Japan
{elboukkouri,lavergne,pz}@limsi.fr, olivier.ferret@cea.fr
{hiroshi.noji,j-tsujii}@aist.go.jp

Abstract

Due to the compelling improvements brought by BERT, many recent representation models adopted the Transformer architecture as their main building block, consequently inheriting the *wordpiece* tokenization system despite it not being intrinsically linked to the notion of Transformers. While this system is thought to achieve a good balance between the flexibility of characters and the efficiency of full words, using predefined wordpiece vocabularies from the general domain is not always suitable, especially when building models for *specialized domains* (e.g., the medical domain). Moreover, adopting a wordpiece tokenization shifts the focus from the word level to the subword level, making the models conceptually more complex and arguably less convenient in practice. For these reasons, we propose CharacterBERT, a new variant of BERT that drops the wordpiece system altogether and uses a Character-CNN module instead to represent *entire words* by consulting their *characters*. We show that this new model improves the performance of BERT on a variety of medical domain tasks while at the same time producing robust, word-level, and open-vocabulary representations.

1 Introduction

Pre-trained language representations from Transformers (Vaswani et al., 2017) have become arguably the most popular choice for building NLP systems[1]. Among all such models, BERT (Devlin et al., 2019) has probably been the most successful, spawning a large number of new improved variants (Liu et al., 2019; Lan et al., 2019; Sun et al., 2019; Zhang et al., 2019; Clark et al., 2020). As a result, many of the recent language representation models inherited BERT's subword tokenization system which relies on a predefined set of *wordpieces* (Wu et al., 2016), supposedly striking a good balance between the flexibility of characters and the efficiency of full words.

While current research mostly focuses on improving language representations for the default "general-domain", there seems to be a growing interest in building suitable word embeddings for more *specialized* domains (El Boukkouri et al., 2019; Si et al., 2019; Elwany et al., 2019). However, with the growing complexity of recent representation models, the default trend seems to favor re-training general-domain models on specialized corpora rather than building models from scratch with a specialized vocabulary (e.g., BlueBERT (Peng et al., 2019) and BioBERT (Lee et al., 2020)). While these methods undeniably produce good models [2], a few questions remain: How suitable are the predefined general-domain vocabularies when used in the context of specialized domains (e.g., the medical domain)? Is it better to train specialized models with specialized subword units? Do we induce any biases by training specialized models with general-domain wordpieces?

In this paper, we propose CharacterBERT, a possible solution for avoiding any biases that may come from the use of a predefined wordpiece vocabulary, and an effort to revert back to conceptually simpler word-level models. This new variant does not rely on wordpieces but instead consults the characters of

[1]See the leaderboard of the GLUE benchmark.
[2]See the baselines from the BLUE benchmark.

Proceedings of the 28th International Conference on Computational Linguistics, pages 6903–6915
Barcelona, Spain (Online), December 8-13, 2020

each token to build representations similarly to previous word-level open-vocabulary systems (Luong and Manning, 2016; Kim et al., 2016; Jozefowicz et al., 2016). In practice, we replace BERT's wordpiece embedding layer with ELMo's (Peters et al., 2018) Character-CNN module while keeping the rest of the architecture untouched. As a result, CharacterBERT is able to produce word-level contextualized representations and does not require a wordpiece vocabulary. Furthermore, this new model seems better suited than vanilla BERT for training specialized models, as evidenced by an evaluation on multiple tasks from the medical domain. Finally, as expected from a character-based system, CharacterBERT is also seemingly more robust to noise and misspellings. To the best of our knowledge, this is the first work that replaces BERT's wordpiece system with a word-level character-based system.

Our contributions are the following:

- We provide preliminary evidence that general-domain wordpiece vocabularies are not suitable for specialized domain applications.
- We propose CharacterBERT, a new variant of BERT that produces word-level contextual representations by consulting characters.
- We evaluate CharacterBERT on multiple specialized medical tasks and show that it outperforms BERT without requiring a wordpiece vocabulary.
- We exhibit signs of improved robustness to noise and misspellings in favor of CharacterBERT.
- We enable the reproducibility of our experiments by sharing our pre-training and fine-tuning codes. Furthermore, we also share our pre-trained representation models to benefit the NLP community[3].

This work has only focused on the English language and the medical (clinical and biomedical) domain. The generalization to other languages and specialized domains is left to future work.

2 General-Domain Wordpieces in Specialized Domains

Since many specialized versions of BERT come from re-training the original model on a set of specialized texts, we carry out a couple of preliminary experiments to gauge the effect of using a general-domain wordpiece vocabulary in a specialized domain. Here we focus on the medical domain for which we learn[4] a new wordpiece vocabulary using MIMIC-III clinical notes (Johnson et al., 2016) and PMC OA[5] biomedical article abstracts. We then process a sample (1M tokens) of the medical corpus with either the medical vocabulary or BERT's original vocabulary and examine the difference.

Looking at the frequency of splitting an unknown token into multiple wordpieces (cf. Figure 1) we see that the medical vocabulary produces overall less wordpieces than the general version, both at occurrence and type levels. Moreover, we see that $\approx 13\%$ of occurrences are never split as they are already part of the medical vocabulary but are decomposed into two or more wordpieces by the general vocabulary.

Reference	**Medical Vocabulary**	**General Vocabulary**
paracetamol	[paracetamol	[para, ce, tam, ol]
choledocholithiasis	[choledoch, olithiasis]	[cho, led, och, oli, thi, asi, s]
borborygmi	[bor, bor, yg, mi]	[bo, rb, ory, gm, i]

Table 1: Comparison of the tokenization of specific medical terms by vocabularies from different domains.

When looking closer at the quality of the produced wordpieces (cf. Table 1), we see that in addition to producing fewer subwords, the specialized vocabulary also seems to produce more meaningful units (e.g. "choledoch" and "olithiasis"). These preliminary analyses show that the choice of a vocabulary

[3]Our models and code are available at: `https://github.com/helboukkouri/character-bert`.

[4]We use the open-source implementation in SentencePiece.

[5]PubMed Central Open Access.

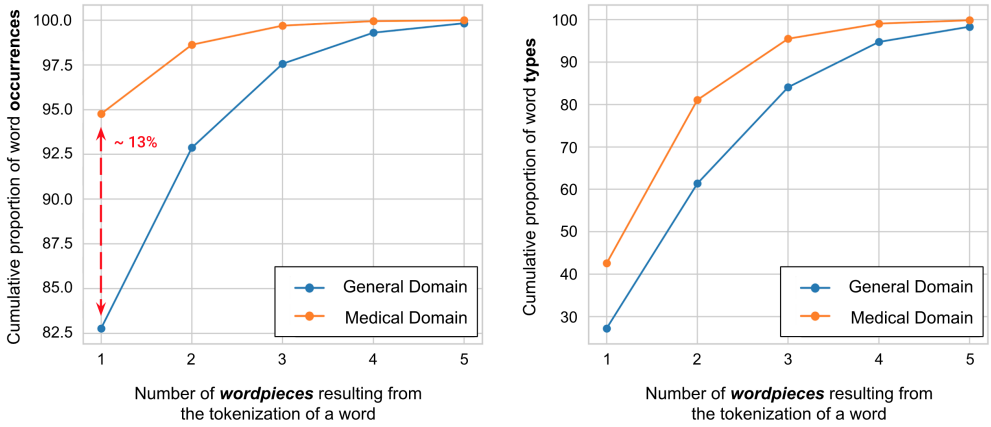

Figure 1: Comparison of the tokenization of a medical corpus by vocabularies from different domains.

affects the quality of the tokenization which may in turn induce biases in downstream applications of the representation model. To avoid such biases, and in an effort to revert back to more convenient and conceptually simpler word-level models, we propose CharacterBERT, a wordpiece-free variant of BERT.

3 CharacterBERT

CharacterBERT is similar in every way to vanilla BERT but uses a different method to construct initial context-independent representations: while the original model consults its vocabulary to split unknown tokens into multiple wordpieces then embeds each unit independently using a wordpiece embedding matrix, CharacterBERT uses a Character-CNN module (Peters et al., 2018; Jozefowicz et al., 2016) which consults the characters of a token to produce a single representation (see Figure 2).

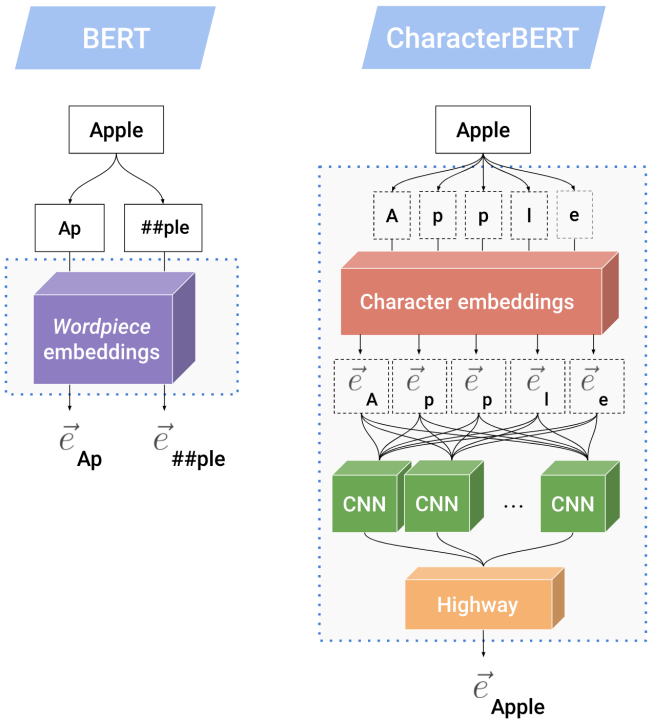

Figure 2: Comparison of the context-independent representation systems in BERT and CharacterBERT. In this illustration, BERT splits the word "Apple" into two wordpieces then embeds each unit separately. CharacterBERT produces a single embedding for "Apple" by consulting its sequence of characters.

3.1 Character-CNN: Building Word Representations From Characters

We use the Character-CNN that is implemented as part of ELMo's architecture. This module constructs context-independent token representations through the following steps:

1. Each token is converted into a sequence of characters[6] with a maximum sequence length of 50.

2. A lookup is performed for each character, producing a sequence of 16-d embeddings.

3. The character embedding sequence is fed to multiple 1-d CNNs (LeCun et al., 1989) with different filters[7]. The output of each CNN is then max-pooled across the character sequence and concatenated with other CNN outputs to produce a single representation.

4. The CNN representation then goes through two Highway layers (Srivastava et al., 2015) that apply non-linearities with residual connections before being projected down to a final embedding size which we chose to be coherent with BERT's 768-dimensional wordpiece representations.

As with BERT, we add the token embedding (here, the Character-CNN representation) to position and segment embeddings before feeding the resulting context-independent representation to several Transformer layers. Since CharacterBERT does not split tokens into wordpieces, each input token is assigned a single final contextual representation by the model.

3.2 Pre-training Procedure

Like BERT, our model is pre-trained on two tasks: a Masked Language Modelling task (MLM) and a Next Sentence Prediction task (NSP). The only difference lies in the MLM task where instead of predicting single wordpieces, we predict entire words. This natural consequence of handling words instead of wordpieces is somewhat related to recent work on Whole Word Masking which has been shown to improve the quality of BERT models[8] (Cui et al., 2019).

4 Experiments

We compare BERT and CharacterBERT on multiple medical tasks to evaluate the impact of using a Character-CNN module instead of wordpieces. In an attempt to dissociate this impact from any other effects that may be related to training models in our own specific settings, we train each CharacterBERT model alongside a BERT counterpart in the exact same conditions.

4.1 Model Settings

We base our models on the "base-uncased" version of BERT, which uses 12 Transformer layers with 12 attention heads and produces 768-d representations from uncased texts. This version has ≈ 109.5M parameters and the corresponding CharacterBERT architecture has ≈ 104.6M parameters. It is interesting to note that using a Character-CNN actually results in a smaller overall model despite using a seemingly complex character module. This is because BERT's wordpiece matrix has $\approx$ 30K $\times$ 768-d vectors while CharacterBERT uses smaller 16-d character embeddings with mostly small-sized CNNs.

We pre-train four different models to simulate the usual situation where BERT is first pre-trained on a general corpus before being re-trained on a set of specialized texts:

$\text{BERT}_{\text{general}}$: a general-domain model obtained by pre-training BERT on a general corpus. It uses the same architecture and wordpiece vocabulary as BERT (base, uncased).

$\text{CharacterBERT}_{\text{general}}$: a general-domain model obtained by training CharacterBERT on a general corpus. Besides the Character-CNN, it uses the same architecture as $\text{BERT}_{\text{general}}$.

[6] In practice, the tokens are encoded in UTF-8 and all characters including non-ascii symbols are converted into bytes. This allows us to keep a small byte vocabulary of size 256 to which we add a few special symbols for a total of 263.

[7] We use seven 1-d CNNs with the following filters: [1, 32], [2, 32], [3, 64], [4, 128], [5, 256], [6, 512] and [7, 1024].

[8] Google updated their repository with Whole Word Masking models that improve over the original BERT.

BERT$_{\text{medical}}$: a medical model obtained by re-training BERT$_{\text{general}}$ on a medical corpus.

CharacterBERT$_{\text{medical}}$: a medical model obtained by re-training CharacterBERT$_{\text{general}}$ on a medical corpus. This is the Character-CNN analog of BERT$_{\text{medical}}$.

4.2 Pre-training Phase

4.2.1 Corpora

The original BERT was pre-trained on English Wikipedia and BooksCorpus (Zhu et al., 2015). Since the latter is not publicly available anymore, we replace it with OpenWebText (Gokaslan and Cohen, 2019) to train our general-domain models. We also build a specialized corpus from MIMIC-III and PMC OA abstracts to train our medical-domain models (see Table 2).

Corpus	Composition	# documents	# tokens
General	Wikipedia (EN)	5.99×10^6	2.14×10^9
	OpenWebText	1.56×10^6	1.28×10^9
Medical	MIMIC-III	2.09×10^6	5.05×10^8
	PMC OA abstracts	2.33×10^6	5.22×10^8

Table 2: Statistics on pre-training corpora.

4.2.2 Pre-training Setup

We train each model using 16 Tesla V100-SXM2-16GB GPUs and following the implementation and parameters in the NVIDIA codebase[9]. Each complete pre-training phase consists of two steps:

Step 1 3,519 updates with a batch size[10] of 8,192 and a learning rate of 6.10^{-3} on sequences of size 128.

Step 2 782 updates with a batch size of 4,096 and a learning rate of 4.10^{-3} on sequences of size 512.

All models are optimized using LAMB (You et al., 2019) with a warm-up rate and weight decay of 0.01.

4.3 Evaluation Phase

4.3.1 Tasks

All models are evaluated on five medical tasks after adding task-specific layers (Devlin et al., 2019).

Medical Entity Recognition We evaluate our models on the i2b2/VA 2010 (Uzuner et al., 2011) clinical concept extraction task which aims to extract three types of medical concepts: PROBLEM (e.g. "headache"), TREATMENT (e.g. "oxycodone"), and TEST (e.g. "MRI").

Natural Language Inference We also evaluate on the clinical natural language inference task MEDNLI (Romanov and Shivade, 2018) that aims to classify sentence pairs into three categories: CONTRADICTION, ENTAILMENT, and NEUTRAL.

Relation Classification For more variety, we also evaluate on two biomedical relation classification tasks: ChemProt (Krallinger et al., 2017) from the BioCreative VI challenge and DDI (Herrero-Zazo et al., 2013) from SemEval 2013 - Task 9.2. The goal of ChemProt is to detect and classify chemical-protein interactions as ACTIVATOR (CPR:3), INHIBITOR (CPR:4), AGONIST (CPR:5), ANTAGONIST (CPR:6), or SUBSTRATE (CPR:9). The goal of DDI is to detect and classify drug-drug interactions into the following categories: ADVISE (DDI-advise), EFFECT (DDI-effect), MECHANISM (DDI-mechanism), and INTERACTION (DDI-int).

[9]More specifically, we adapt these scripts to our needs.
[10]We use gradient accumulation for larger batch sizes.

Sentence Similarity Finally, we also evaluate our models on the clinical sentence similarity task ClinicalSTS (Wang et al., 2018a) from BioCreative/OHNLP Challenge 2018, Task 2 (Wang et al., 2018b). The goal here is to produce similarity scores for sentence pairs that correlate with the gold standard.

We provide examples for each task in Figure 3 and report the number of examples in Table 3.

	i2b2	MEDNLI	ChemProt	DDI	ClinicalSTS
Train	24,757	11,232	19,460	18,779	600
Val.	6,189	1,395	11,820	7,244	150
Test	45,404	1,422	16,943	5,761	318

Table 3: Number of examples of each evaluation task.

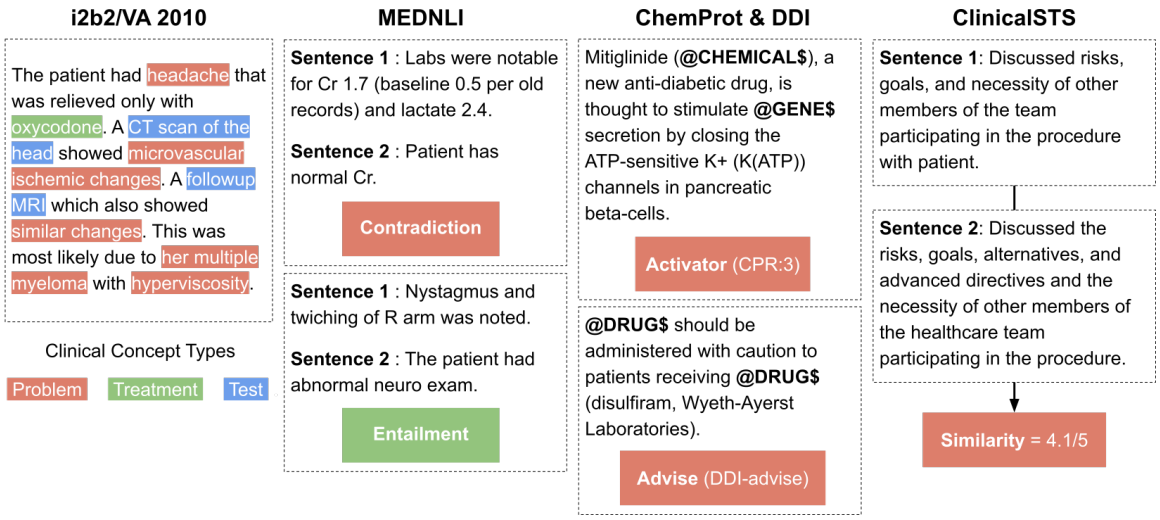

Figure 3: Examples from each evaluation task.

4.3.2 Evaluation Setup

Given all the pre-trained models, the evaluation tasks, and a set of random seeds $i \in 1..10$:

1. We choose a pre-trained model, an evaluation task, and a random seed i then run 15 training epochs with batches of size 32.

2. At each epoch, we evaluate the model on a validation set that is either given or computed as 20% of the training set. According to the validation performance, we save the best model.

3. After completing all training epochs, we load the best model and evaluate it on the test set.

4. We repeat the whole process for all seeds to compute a final performance as *mean* $\pm$ *std*.

In addition to being useful for measuring model variability, fine-tuning 10 versions for each model also enables us to build ensembles. In fact, by using a majority voting strategy, we are able to combine the predictions from each seed into a single ensemble model[11]. In practice we do not use all seeds at once: we exclude a single seed, build an ensemble then repeat this process to get 10 ensembles for each model setting which can be used to compute a final ensemble performance as *mean* $\pm$ *std*. All fine-tuning experiments are run on a single Tesla V100-PCIE-32GB and are optimized using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 3e-5, a warm-up ratio of 0.1, and a weight decay of 0.1.

[11]For ClinicalSTS, we use the average predicted score instead of a majority class since the targets are continuous.

5 Results and Discussion

5.1 Speed Benchmark

5.1.1 Pre-training

Using the setup detailed in Section 4.2.2, training a single BERT through Steps 1 and 2 takes around 26.5 hours for BERT and 55 hours for CharacterBERT even though both architectures have about the same number of parameters. This large gap in pre-training speed is partly due to the Character-CNN being slower to train as it is more complex than the original wordpiece embedding matrix. However, the main reason for the slower pre-training is that we are not able to use a very specific trick during Masked Language Modelling. In fact, BERT shares the parameters of its wordpiece embedding matrix with the MLM output layer, which allows it to train faster. In our case, since we do not use wordpieces, we build a temporary vocabulary from the top 100K tokens in the training corpus and use them as targets for MLM[12]. We expect that improved pre-training speed can be achieved using Noise Contrastive Estimation (Mnih and Kavukcuoglu, 2013) or similar methods. However, such optimizations are left for future work.

5.1.2 Fine-tuning

In addition to pre-training speed, we also report the fine-tuning speed both at training and inference time.

Fine-tuning (w/ Tesla V100-PCIE-32GB)

	i2b2	MEDNLI	STS	DDI	ChemProt
BERT	3:36:20	1:09:29	0:02:58	1:32:42	2:42:36
CharacterBERT	4:29:01	1:22:40	0:04:12	1:19:43	3:25:31
Relative difference	+24.35%	+18.97%	+41.57%	-14.01%	+26.39%

Inference (w/ Tesla V100-PCIE-32GB)

	i2b2	MEDNLI	STS	DDI	ChemProt
BERT	0:11:16	0:00:11	0:00:01	0:00:31	0:02:28
CharacterBERT	0:10:57	0:00:10	0:00:01	0:00:22	0:02:44
Relative difference	-2.81%	-9.09%	0.00%	-29.03%	+10.81%

Figure 4: Training/inference speed comparison.

Figure 4 shows that CharacterBERT is much less at a disadvantage when it comes to fine-tuning (19% slower on average instead of 108%). However, in the specific case of the DDI task, CharacterBERT is actually 14% faster than BERT. This exceptional behavior may be due to the presence of many domain-specific terms that are split into multiple wordpieces, thus increasing the input size with BERT. In fact, since our model works at the word level, the input size is stable and data batches may be processed faster than with BERT. At inference time, CharacterBERT is slightly faster than BERT as the Character-CNN is not as slow during inference as it is during optimization.

5.2 Reproducing Vanilla Models

We report the performance of BERT(base, uncased) as well as BlueBERT(base, uncased) (Peng et al., 2019), a medical model pre-trained on MIMIC-III and PubMed abstracts[13]. Including these results allows us to evaluate the quality of our pre-training procedure. Figure 5 shows that $\text{BERT}_{\text{general}}$ performs slightly worse than the original BERT despite using exactly the same architecture. However, this difference is small and can be attributed to either the different general-domain corpora (OpenWebText instead of BooksCorpus) or to differences in pre-training parameters (number of updates, batch size...). Moreover, we see that $\text{BERT}_{\text{medical}}$ performs at the same level as BlueBERT, sometimes outperforming the latter substantially ($\approx +4$ F1 on ChemProt), which allows us to safely assume our pre-training procedure to be correct.

5.3 Ensembles and Model Selection

We can see from Figure 5 that ensembles (orange bars) clearly improve over single models (blue bars). While not surprising *per se*, it is worth noting that these ensembles were produced using a naive majority voting strategy which can easily be applied as a post-processing step. Moreover, we see that the test

[12]Please note that this also means that we never mask tokens that are not within the top 100K most frequent tokens.

[13]Note that BlueBERT is trained on PubMed abstracts while our medical models are trained on PMC OA abstracts.

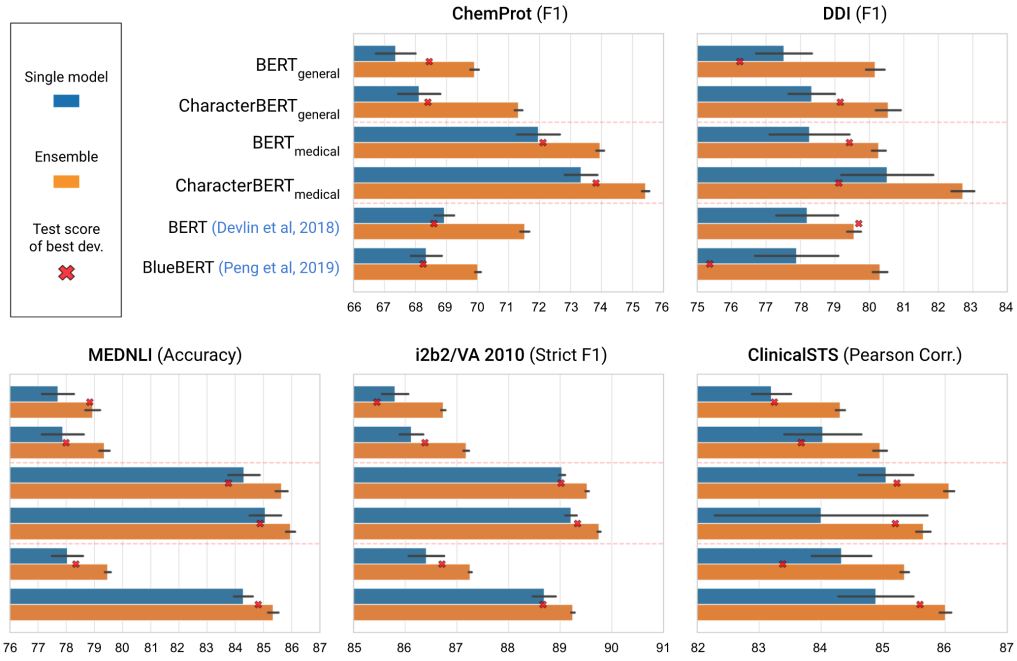

Figure 5: Comparison of pre-trained models when fine-tuned on several medical tasks. For each model, the test performance of 10 random seeds is expressed as *mean* $\pm$ *std* and is shown in blue for single models and orange for ensembles. The performance of the best validation seed is shown in red.

results of the best validation model (red symbol) are always below those of the ensembles' performance. Finally, we note that ensembles have substantially lower variances compared to single models, which makes them more reliable for comparisons.

5.4 BERT vs. CharacterBERT: How Significant Is the Difference?

Figure 5 shows that CharacterBERT often improves over a vanilla BERT. In particular, our medical model improves over the ensemble performance of $\text{BERT}_{\text{medical}}$ by ≈ 1.5 points on ChemProt, ≈ 2 points on DDI, and ≈ 0.5 points on MEDNLI and i2b2. However, we see that $\text{CharacterBERT}_{\text{medical}}$ performs worse than BERT in the specific case of ClinicalSTS and suffers from a surprisingly high variance. Since the ClinicalSTS dataset is also very small compared to the other datasets, these results should be taken with care even if the difference with BERT seems to be significant according to Figure 6. Results with general-domain models seem to also be in favor of CharacterBERT. However, these differences may not be substantial.

To provide a more rigorous evaluation of the statistical significance of our results, we perform Almost Stochastic Order tests (ASO) (Dror et al., 2019) for each pair of models. ASO tests aim to determine whether a stochastic order exists between two models based on their respective sets of evaluation scores. In practice, given the 10 single model scores of two chosen models A and B, the method computes a test-specific value ϵ that indicates how far model A is from being significantly better than model B. This distance ϵ is equal to 0 when model A $\succeq$ B, 1 when B $\succeq$ A, and 0.5 when no order can be determined. Figure 6 shows the values of ϵ for all model pairs on each task. Looking at the average significance matrix, we can see that $\text{CharacterBERT}_{\text{general}}$ improves over its BERT counterpart (cell [d,c]). Moreover, we see that the overall best model is $\text{CharacterBERT}_{\text{medical}}$ as evidenced by the bottom blue row (cells [f,a] to [f,e]), which further validates that our model indeed improves over vanilla BERT.

5.5 Robustness to Noise and Misspellings

We want to investigate whether CharacterBERT is more robust to noise and misspellings than BERT. For that purpose, we create noisy versions of the MEDNLI corpus where, given a noise level of X%, we transform each token with the same probability into a misspelled version either by removing, adding,

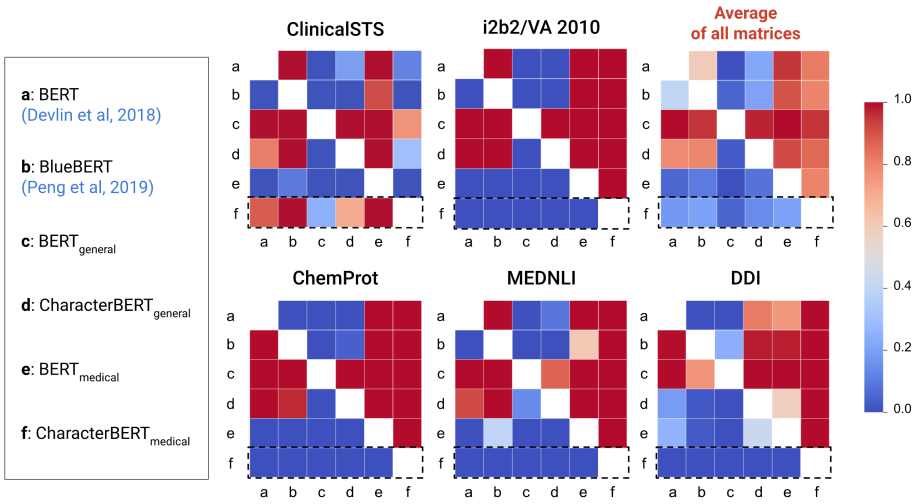

Figure 6: Statistical significance: Minimal distance ϵ for Almost Stochastic Order at level $\alpha = 5\%$. Blue cells mean that the left model is significantly better than the bottom model. Red cells mean the opposite.

replacing a single character or swapping two consecutive characters. We conduct experiments where noise is added to the test set only as well as experiments adding noise to the train/dev/test splits.

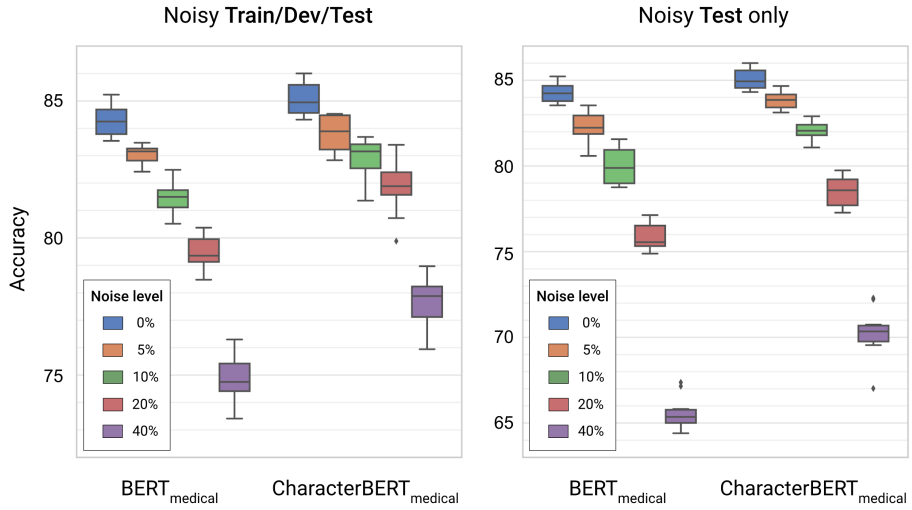

Figure 7: Comparing BERT and CharacterBERT on noisy (misspelled) versions of MEDNLI test.

Figure 7 shows the results for $\text{BERT}_{\text{medical}}$ and $\text{CharacterBERT}_{\text{medical}}$ with various noise levels. We see that the latter is indeed more robust to misspellings as evidenced by the slower decrease in performance compared to BERT. In particular, when a noise level of 40% is applied to the test set only, CharacterBERT is $\approx$ 5 F1 higher than BERT whereas the original difference between the two models was $<$ 1 F1. Experiments adding noise to all splits show that both models can learn to be more robust, however, CharacterBERT remains at an advantage.

5.6 Discussion and Future Work

Overall CharacterBERT seems to either perform at the same level or improve over BERT. This is especially true for the specialized versions and is further validated by the ASO tests. The new variant also seems to be more robust to misspellings while at the same time producing word-level open-vocabulary representations. This improved robustness is desirable since BERT seems to be sensitive to misspellings (Pruthi et al., 2019; Sun et al., 2020). On the downside, CharacterBERT is slower to pre-train, although not as slow to

fine-tune and even slightly faster at inference time. Future work may apply a Character-CNN to recent Transformer-based models (Lan et al., 2019; Sun et al., 2019), optimize the pre-training architecture to improve its speed, or explore any other advantages of a character-level system over wordpieces.

6 Conclusion

The overall strategy when building specialized versions of BERT seems to be re-training the original model on a specialized corpus. This implies keeping a general-domain wordpiece vocabulary that may not be suited for the domain of interest. Our main contribution is CharacterBERT, a variant of BERT that drops the wordpiece system altogether in favor of a Character-CNN. This module represents tokens by consulting their characters, allowing our model to produce word-level open-vocabulary representations. We evaluate CharacterBERT and show that it globally outperforms BERT when specialized for the medical domain while at the same time being more robust to misspellings.

Acknowledgments

This work has been funded by the French National Research Agency (ANR) and is under the ADDICTE project (ANR-17-CE23-0001). Moreover, computational resource of AI Bridging Cloud Infrastructure (ABCI)[14] provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

References

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese BERT. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy, July. Association for Computational Linguistics.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, and Pierre Zweigenbaum. 2019. Embedding strategies for specialized domains: Application to clinical entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 295–301, Florence, Italy, July. Association for Computational Linguistics.

Emad Elwany, Dave Moore, and Gaurav Oberoi. 2019. Bert goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding. *arXiv preprint arXiv:1911.00473*.

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

[14] `https://abci.ai/`

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, et al. 2017. Overview of the BioCreative VI chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany, August. Association for Computational Linguistics.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*, pages 2265–2273.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. 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. Association for Computational Linguistics.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium, October-November. Association for Computational Linguistics.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. ERNIE 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-BERT: BERT is not robust on misspellings! generating nature adversarial samples on BERT. *arXiv preprint arXiv:2003.04985*.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yanshan Wang, Naveed Afzal, Sunyang Fu, Liwei Wang, Feichen Shen, Majid Rastegar-Mojarad, and Hongfang Liu. 2018a. MedSTS: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, pages 1–16.

Yanshan Wang, Naveed Afzal, Sijia Liu, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Sunyang Fu, and Hongfang Liu. 2018b. Overview of the BioCreative/OHNLP challenge 2018 task 2: clinical semantic textual similarity. *Proceedings of the BioCreative/OHNLP Challenge*, 2018.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2019. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Appendices

A.1 Detailed Test Scores

Figure 8 provides numerical evaluation scores for the models displayed in Figure 5. To give a better idea about the distribution of model scores, these are reported as first, second (median), and third quartiles.

		ChemProt (F1 score)			DDI (F1 score)			MEDNLI (Accuracy)			i2b2/VA 2010 (Strict F1 score)			ClinicalSTS (Pearson Correlation)		
		Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
$BERT_{general}$	S	66.94	67.04	67.84	76.93	77.40	77.99	77.43	77.67	77.94	85.72	85.86	85.97	83.07	83.22	83.35
	E	69.81	69.88	69.98	79.99	80.14	80.35	78.71	78.90	79.15	86.71	86.73	86.77	84.27	84.33	84.36
$CharacterBERT_{general}$	S	67.89	68.24	68.38	77.89	78.17	79.03	77.18	78.09	78.50	85.99	86.18	86.28	83.63	83.91	84.09
	E	71.27	71.30	71.40	80.39	80.54	80.88	79.20	79.29	79.47	87.15	87.17	87.23	84.94	84.97	85.03
$BERT_{medical}$	S	71.76	71.93	72.23	77.54	77.93	79.15	83.79	84.25	84.69	88.99	89.01	89.08	84.80	84.98	85.20
	E	73.85	73.94	74.01	80.14	80.20	80.38	85.51	85.65	85.78	89.49	89.51	89.55	86.01	86.08	86.12
$CharacterBERT_{medical}$	S	72.84	73.44	73.78	79.18	80.38	81.70	84.56	84.95	85.58	89.14	89.24	89.30	82.92	84.80	85.15
	E	75.31	75.40	75.50	82.44	82.74	83.01	85.83	85.97	86.11	89.73	89.75	89.77	85.54	85.62	85.76
BERT (Devlin et al, 2018)	S	68.67	68.82	69.18	77.67	78.08	78.83	77.67	78.02	78.29	86.23	86.54	86.61	83.97	84.44	84.65
	E	71.46	71.54	71.64	79.46	79.49	79.61	79.41	79.47	79.54	87.23	87.26	87.28	85.32	85.37	85.40
BlueBERT (Peng et al, 2019)	S	68.25	68.31	68.69	77.55	77.89	78.74	84.07	84.25	84.55	88.47	88.73	88.87	84.39	84.98	85.39
	E	69.93	69.98	70.10	80.26	80.33	80.43	85.25	85.41	85.44	89.22	89.24	89.28	85.95	85.99	86.06

Figure 8: Performance of our pre-trained models when fine-tuned on five different medical tasks. Two baselines are included: BERT (Devlin et al., 2019) using the "base-uncased" architecture, and BlueBERT (Peng et al., 2019) a medical BERT that is the result of re-training the former model on MIMIC-III and PubMed abstracts. Legend: Qi = i-th quartile, E = ensemble, S = single model.