Modular Adaptation of Multilingual Encoders to Written Swiss German Dialect

Jannis Vamvas   **Noëmi Aepli**   **Rico Sennrich**

Department of Computational Linguistics, University of Zurich
`{vamvas,naepli,sennrich}@cl.uzh.ch`

Abstract

Creating neural text encoders for written Swiss German is challenging due to a dearth of training data combined with dialectal variation. In this paper, we build on several existing multilingual encoders and adapt them to Swiss German using continued pre-training. Evaluation on three diverse downstream tasks shows that simply adding a Swiss German adapter to a modular encoder achieves 97.5% of fully monolithic adaptation performance. We further find that for the task of retrieving Swiss German sentences given Standard German queries, adapting a character-level model is more effective than the other adaptation strategies. We release our code and the models trained for our experiments.[1]

1 Introduction

When applying natural language processing (NLP) techniques to languages with dialectal variation, two typical challenges are a lack of public training data as well as varying spelling conventions. In the case of Swiss German, which is spoken by around 5 million people and is often used for informal written communication in Switzerland, these factors make it more challenging to train a BERT-like text encoder for written text.

In this paper, we adapt pre-trained multilingual encoders to Swiss German using continued pre-training on a modest amount of Swiss German training data. We evaluate the approaches on part-of-speech (POS) tagging with zero-shot cross-lingual transfer from Standard German (Aepli and Sennrich, 2022), as well as dialect identification (Zampieri et al., 2019) and cross-lingual sentence retrieval based on a parallel Standard German–Swiss German test set (Aepli et al., 2023).

We find that depending on the multilingual encoder, continued pre-training leads to an average

	Monolithic	Modular
Subwords	XLM-R → Swiss German XLM-R	X-MOD/SwissBERT → Swiss German adapter
Characters	CANINE → Swiss German CANINE	X-MOD/SwissBERT → Swiss German character-level adapter

Table 1: Overview of the encoder models we release.

improvement of 10%–45% in average accuracy across the three downstream tasks. We then focus on comparing monolithic adaptation, where all the parameters of the encoder are updated during continued pre-training, to modular adaptation with language-specific modular components (*language adapters*; Pfeiffer et al., 2022). Even though modular adaptation only updates a fraction of the parameters, it is competitive to monolithic adaptation. Given these findings, we propose to extend the SwissBERT model (Vamvas et al., 2023), which was trained on Standard German and other languages, with a Swiss German adapter (Table 1).

We further hypothesize that the architecture of CANINE (Clark et al., 2022), a tokenization-free model that operates on characters, might be better suited to the highly variable spelling of Swiss German. Indeed, a CANINE model adapted to Swiss German excels on the retrieval tasks, while POS tagging works better with subwords.

Finally, we aim to combine the best of both worlds by integrating character-level down- and upsampling modules into a subword-based model and training a *character-level adapter* for Swiss German. However, this jointly modular and tokenization-free strategy underperforms the individual approaches. We hope that our findings can inform the development of modular approaches for other languages with dialectal variation.

---

[1] `https://github.com/ZurichNLP/swiss-german-text-encoders`

2 Adaptation Scenario

Our goal is to train an encoder model for Swiss German (language code gsw) with limited training data. Since Standard German (language code de) is a closely related language, we focus on transfer learning from Standard German to Swiss German. We rely on pre-trained multilingual models that have already been trained on Standard German, and adapt them to Swiss German using continued pre-training.

Swiss German adaptation data For training on Swiss German, we use the SwissCrawl corpus (Linder et al., 2020), which contains 11M tokens of Swiss German text extracted from the web. The text in SwissCrawl exhibits some normalizations that eventual input text will not have, e.g., isolation of individual sentences, normalization of punctuation and emoji removal. To diversify the training data, we extend the pre-training dataset with a custom collection of 382k Swiss German tweets. In total, we use 18M tokens for pre-training on Swiss German. Both datasets were automatically mined and may contain some text in other languages.

Standard German data To promote transfer from Standard German to Swiss German later on, we include an equal part of Standard German data in the continued pre-training data. We use a sample of news articles retrieved from the Swissdox@LiRI database, comparable to the data the SwissBERT model has been trained on (Vamvas et al., 2023).

3 Monolithic Approaches

We evaluate a subword-based model and a character-based model, with and without continued pre-training on Swiss German. We call these models monolithic (non-modular), because the entire model is updated during continued pre-training.

3.1 XLM-R

We train XLM-R (Conneau et al., 2020) with masked language modeling (MLM). XLM-R was pre-trained on 100 languages, which include Standard German but not Swiss German.

3.2 CANINE

The CANINE model (Clark et al., 2022) was pre-trained on 104 languages, again including Standard German but excluding Swiss German. Unlike XLM-R, CANINE directly encodes character

sequences and does not require a tokenizer at inference time. This is achieved by extending the standard transformer architecture with character down- and upsampling modules.

The *downsampling module* combines a single-layer blockwise transformer with strided convolution, which reduces the sequence length by a factor of $r = 4$, where r is a hyperparameter. As a consequence, the standard transformer does not see every character individually, but only sees downsampled positions. The *upsampling module*, which is needed for token-level tasks, mirrors the downsampling procedure and restores the original sequence length. We refer to Clark et al. (2022) for a detailed description of the architecture.

Clark et al. (2022) describe two alternative approaches for pre-training: CANINE-S, which uses a tokenizer to determine masked tokens and is similar to standard MLM, and CANINE-C, which is an autoregressive character loss. In our experiments, we use CANINE-S with the SwissBERT subword tokenizer to perform continued pre-training.

4 Modular Approaches

4.1 SwissBERT

We base our adapter experiments on SwissBERT (Vamvas et al., 2023), a variant of X-MOD (Pfeiffer et al., 2022) that includes language adapters for Standard German, French, Italian and Romansh. Compared to the original X-MOD model, which was trained with language adapters for 81 languages, SwissBERT has a custom SentencePiece vocabulary and word embeddings optimized for Switzerland-related text, and we assume that this is beneficial for continued pre-training on Swiss German.

4.2 Subword-level Adapter for SwissBERT

We add a Swiss German adapter to SwissBERT and freeze the parameters of the model except for the adapter modules during continued pre-training. We initialize the Swiss German adapter with the weights of the Standard German adapter and pre-train it on the Swiss German part of our dataset. During fine-tuning on downstream tasks, we freeze the adapters and update the remainder of the model.

For this approach, we only use the Swiss German part of our pre-training corpus for continued pre-training, and not Standard German, since the modular architecture is expected to allow for cross-lingual transfer without continued pre-training

	POS	GDI	Retrieval		Macro-Avg.
			GSW-BE	GSW-ZH	
XLM-R:					
– without continued pre-training	52.6±1.8	47.2±15.1	60.6	75.7	56.0
– with continued pre-training	<u>86.9±0.3</u>	62.1±0.8	91.1	96.0	<u>80.9</u>
CANINE:					
– without continued pre-training	46.7±1.3	59.0±0.6	92.8	94.8	66.5
– with continued pre-training	60.9±1.4	60.8±0.4	<u>96.4</u>	<u>96.9</u>	72.8
SwissBERT:					
– DE adapter without continued pre-training	64.8±2.0	61.3±0.5	66.1	82.2	66.7
– subword-level GSW adapter	83.2±0.3	62.0±0.4	82.9	92.4	77.6
– character-level GSW adapter	41.5±0.9	51.9±1.3	35.6	42.6	44.2

Table 2: Comparison of different models on three downstream tasks: part-of-speech (POS) tagging accuracy, German dialect identification (GDI) F1-score, and cross-lingual sentence retrieval accuracy. For the supervised tasks, we report the average and standard deviation across 5 fine-tuning runs. Underlined results indicate the best performance for a task.

on the source language. Table A4 provides an overview of the languages used for each approach.

4.3 Character-level Adapter for SwissBERT

Previous work has found that learning a custom subword segmentation and embeddings that are adapted to the vocabulary of the target language can improve performance (Wang et al., 2019; Pfeiffer et al., 2021; Vamvas et al., 2023). However, this limits the degree of modularity, and we thus investigate a tokenization-free approach as an alternative. In this experiment, we discard SwissBERT's subword embeddings when training the Swiss German adapter, and instead add the downsampling and upsampling modules of the CANINE architecture.[2]

Adding these modules results in exactly the same architecture as CANINE, except that we opt for byte embeddings instead of character hash embeddings. CANINE uses a hash embedding method that can map any Unicode code point to a fixed-size embedding. Since Standard German and Swiss German are mainly written in Latin script and there are limited training data, we forgo the hash embedding and learn UTF-8 byte embeddings instead.

Using the CANINE-S objective, we first pre-train the character modules on Standard German pre-training data. We then continue pre-training the adapters and the joint character modules on both languages, while freezing the rest of the model. During fine-tuning, we freeze the adapters and train

the remainder, analogous to the subword-level experiment.

5 Evaluation

5.1 Part-of-Speech Tagging (POS)

Following Aepli and Sennrich (2022), we evaluate our models on POS tagging with zero-shot cross-lingual transfer from Standard German. To train the models, we use the German HDT Universal Dependencies Treebank (Borges Völker et al., 2019) and test on a dataset introduced by Hollenstein and Aepli (2014). We report accuracy across the 54 STTS tags (Schiller et al., 1999).[3] We rely on the provided word segmentation and label the first token (subword/character/byte) of each word.

5.2 German Dialect Identification (GDI)

The GDI task (Zampieri et al., 2019) is based on transcripts of the ArchiMob corpus of spoken Swiss German (Samardžić et al., 2016). This dataset contains four dialects, namely, Bern, Basel, Lucerne, and Zurich regions, constituting four distinct classes. We report the weighted F1-score.

5.3 Sentence Retrieval

For evaluating cross-lingual sentence retrieval, we use human translations of the English newstest2019 source dataset (Barrault et al., 2019) into different languages. Translations into

---

[2]We term this approach GLOBI (**G**ranular **Lo**calization of **Bi**directional Encoders).

[3]We mask the APPRART gold tag, which is not included in the training tag set, when calculating accuracy.

	POS	GDI	Retrieval		Macro-Avg.
			GSW-BE	GSW-ZH	
SwissBERT subword-level GSW adapter:					
– only updating the adapter weights	83.2±0.3	62.0±0.4	82.9	92.4	77.6 (97.5%)
– also updating the word embeddings	83.9±0.1	62.1±0.3	86.0	93.7	78.6 (98.7%)
– updating all the weights	85.7±0.3	63.1±0.3	86.6	93.4	79.6 (100%)

Table 3: Effect of modularity on continued pre-training: Only updating the adapter weights during continued pre-training achieves 97.5% of the accuracy of a monolithic baseline where we update all the parameters of SwissBERT.

Standard German are provided by NTREX-128 (Federmann et al., 2022); translations into Swiss German are provided by Aepli et al. (2023) for two regions, Bern (gsw-be) and Zurich (gsw-zh).

For both Swiss German test sets, we report the top-1 accuracy of retrieving the correct translation among all 1,997 translations, given the Standard German equivalent. Note that 100% accuracy is not attainable, since newstest2019 has a small number of duplicate or near-duplicate sentences. Following an evaluation approach used for SwissBERT (Vamvas et al., 2023), we perform unsupervised retrieval with the BERTScore metric (Zhang et al., 2020). We average the hidden states across all encoder layers. In the case of the CANINE-style models, we use only the transformer layers that represent the downsampled positions.

6 Experimental Setup

Continued pre-training We combine Swiss German and Standard German training data with a 1:1 ratio. The resulting bilingual dataset contains 37M tokens in total, and we set aside 5% for validation (Table A6). We set the learning rate to 1e-4 and select the best checkpoint based on the validation loss out of 10 epochs; otherwise we use the default settings of Hugging Face transformer's MLM example script. We train the models on a Nvidia V100 GPU with 32GB of memory and adjust the batch size dynamically to fit the available memory. With the subword-based models, we set the sequence length to 512. With the CANINE-style models, we use the default downsampling rate of $r = 4$ and a sequence length of $r \times 512 = 2048$ tokens (characters or bytes).

Fine-tuning For the downstream tasks that involve fine-tuning (POS and GDI), we fine-tune the model with a learning rate of 2e-5 and a batch size of 16. We train for 10 epochs and select the best checkpoint based on the validation accuracy. We report average and standard deviation across 5 fine-tuning runs with different random seeds.

7 Results

Table 2 presents a comparison of the different models on the three downstream tasks. Continued pre-training is highly beneficial for written Swiss German, confirming previous work (Muller et al., 2021; Aepli and Sennrich, 2022; Aepli et al., 2023). This finding extends to the CANINE model, for which language-adaptive pre-training has not been tested before, to our knowledge.

The adapted CANINE shows state-of-the-art performance on the retrieval tasks. A simple ChrF baseline (Popović, 2015) achieves only 90.9% and 93.0% accuracy on the two retrieval tasks, and both the original and the adapted CANINE clearly surpass this baseline. However, the CANINE model has low accuracy on POS tagging, reflecting previous findings for named entity recognition (Clark et al., 2022). Future work could explore alternative strategies for token-level classification tasks.

While the monolithic XLM-R model performs best overall, we consider adding a subword-based Swiss German adapter to SwissBERT a competitive alternative, with the number of trainable parameters reduced by 95% (see Table A1 for a comparison of the model sizes). Table 3 confirms that restricting the continued pre-training to the adapter weights conserves most of the accuracy, compared to updating all the parameters of SwissBERT.

Finally, a character-level adapter, where character up- and downsampling modules are added to the model specifically for Swiss German, performs better than random but clearly worse than the standard approaches. This indicates that while the transformer layers of a subword-based model bear some similarity to the downsampled positions in the CANINE architecture, continued pre-training cannot completely bridge the gap between the two

architectures. Future work could pre-train a modular character-level model from scratch to further improve adaptability to new languages and dialects, while taking into account more recent findings regarding the optimal design of character-level modules for text encoding (Tay et al., 2022; Cao, 2023).

8 Conclusion

We compared strategies for adapting multilingual encoders to Swiss German. We found that the monolithic approach of continued pre-training XLM-R is a strong baseline. Adding a Swiss German adapter to SwissBERT, a model with a modular architecture, is a viable alternative. Finally, adapting CANINE on Swiss German works well for cross-lingual retrieval. The four Swiss German encoder models we trained for our experiments will be made available to the research community.

Limitations

Differences between the pre-trained models make a fair comparison more difficult. The encoder models we compare have originally been pre-trained with different data and hyperparameters (but never on Swiss German). They also differ in their number of parameters and vocabulary sizes, as detailed in Table A1. Furthermore, we use a single, standard set of hyperparameters for pre-training and for evaluation, respectively. Optimizing these hyperparameters for each model individually could lead to further improvements.

Finally, the evaluation results show that it is challenging to perform GDI classification purely based on written text, as previously discussed by Zampieri et al. (2017). In interpreting the results, we focus mainly on the other two tasks, but still report results for GDI to provide a complete picture.

Acknowledgements

References

Noëmi Aepli, Chantal Amrhein, Florian Schottmann, and Rico Sennrich. 2023. A benchmark for evaluating machine translation metrics on dialects without standard orthography. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1045–1065, Singapore. Association for Computational Linguistics.

Noëmi Aepli and Rico Sennrich. 2022. Improving zero-shot cross-lingual transfer between closely related languages by injecting character-level noise. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4074–4083, Dublin, Ireland. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Kris Cao. 2023. What is the best recipe for character-level encoder-only modelling? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5924–5938, Toronto, Canada. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

Nora Hollenstein and Noëmi Aepli. 2014. Compilation of a Swiss German dialect corpus and its application to PoS tagging. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 85–94, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Lucy Linder, Michael Jungo, Jean Hennebert, Claudiu Cristian Musat, and Andreas Fischer. 2020. Automatic creation of text corpora for low-resource languages from the Internet: The case of Swiss German. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2706–2711, Marseille, France. European Language Resources Association.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2016. ArchiMob - a corpus of spoken Swiss German. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4061–4066, Portorož, Slovenia. European Language Resources Association (ELRA).

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textkorpora mit STTS.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*.

Jannis Vamvas, Johannes Graën, and Rico Sennrich. 2023. SwissBERT: The multilingual language model for Switzerland. In *Proceedings of the 8th edition of the Swiss Text Analytics Conference*, pages 54–69, Neuchatel, Switzerland. Association for Computational Linguistics.

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, and Dong Yu. 2019. Improving pre-trained multilingual model with vocabulary expansion. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 316–327, Hong Kong, China. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial evaluation campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15, Valencia, Spain. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei M. Butnaru, and Tommi Jauhiainen. 2019. A report on the third VarDial evaluation campaign. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–16, Ann Arbor, Michigan. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

A List of Encoder Models

Model	Total parameters	Trained	Vocabulary size	URLs (original→adapted)
XLM-R	278M	278M	250,002	⤷ → ⤷
CANINE	132M[†]	132M	-	⤷ → ⤷
SwissBERT				
– subword-level adaptation	139M[‡]	8M	50,262	⤷ → ⤷
– character-level adaptation	123M[‡]	38M[‡]	261	⤷ → ⤷

Table A1: The main encoders trained in this work. [†] Figure does not include the CANINE-S output embeddings, which can be discarded after pre-training. [‡] Figure includes two adapters (Swiss German and Standard German).

B Ablation Study: Custom Subword Vocabulary

	POS	GDI	Retrieval		Macro-Avg.
			GSW-BE	GSW-ZH	
XLM-R:					
– XLM-R vocabulary	86.9±0.3	62.1±0.8	91.1	96.0	80.9
– custom GSW vocabulary	60.3±0.4	60.0±0.6	64.2	79.9	64.1
SwissBERT subword-level GSW adapter[†]:					
– SwissBERT vocabulary	83.9±0.1	62.1±0.3	86.0	93.7	78.6
– custom GSW vocabulary	23.7±2.3	56.9±0.6	65.6	77.3	50.7
CANINE:					
– CANINE-S with SwissBERT vocabulary	60.9±1.4	60.8±0.4	96.4	96.9	72.8
– CANINE-S with custom GSW vocabulary	57.8±1.2	62.1±0.6	95.6	96.3	71.9
SwissBERT character-level GSW adapter:					
– CANINE-S with SwissBERT vocabulary	41.5±0.9	51.9±1.3	35.6	42.6	44.2
– CANINE-S with custom GSW vocabulary	40.6±1.2	11.0±1.9	28.7	38.4	28.4

Table A2: In an ablation experiment, we create a custom subword vocabulary for our continued pre-training dataset using SentencePiece (Kudo and Richardson, 2018). For the subword-based models, we train a new embedding matrix while initializing it with lexically overlapping embeddings from the original model. Using the custom vocabulary for Swiss German decreases performance on all downstream tasks, probably due to the limited amount of training data. For the character-based models, we use the CANINE-S objective with the custom vocabulary. Surprisingly, the custom vocabulary decreases performance, possibly because it is less similar to the subword vocabulary originally used by Clark et al. (2022) to train CANINE-S. [†] In this experiment, we update the embedding weights of SwissBERT to enable a fair comparison.

Vocabulary	Vocabulary Size	Compression Ratio
XLM-R vocabulary	250,002	3.36
SwissBERT vocabulary	50,262	3.37
Custom GSW vocabulary	50,262	4.17

Table A3: Comparison of the SentencePiece vocabularies involved in the above ablation study. We report the compression ratio as the number of characters per subword token in a tokenized sample of our continued pre-training dataset.

C   Model Training Details

Approach	Languages trained	Training samples per second
XLM-R continued pre-training	GSW + DE	88.9
CANINE continued pre-training	GSW + DE	149.6
SwissBERT character-level adapter	GSW + DE	127.1
SwissBERT subword-level adapter:		
– only updating the adapter weights	GSW	215.3
– also updating the word embeddings	GSW	202.4
– updating all the weights	GSW	225.9

Table A4: Empirical training speed in terms of training samples per second. Note that training speed is only comparable for models trained on the same languages, since the DE samples are longer than the GSW samples.

D   Pre-training Datasets

Dataset	Language	Time Range	Examples	Tokens	URL
SwissCrawl (Linder et al., 2020)	GSW	until 2019	563,037	10,961,075	↗
Swiss German Tweets	GSW	2007–2018	381,654	7,259,477	-
Swissdox Sample	DE	2021	409,572	351,643,710	↗

Table A5: Details of the datasets from which we source data for continued pre-training.

Split	Examples (news articles / tweets / sentences)	Tokens
Training GSW	897,477	17,308,288
Training DE	20,140	17,459,689
Validation GSW	47,214	912,264
Validation DE	1,082	905,476

Table A6: Training and validation splits used for continued pre-training.

E   Evaluation Datasets

Dataset	Examples	Tokens	Citation	URL
POS DE (train)	75,617	13,655,973	Borges Völker et al. (2019)	↗
POS DE (validation)	18,434	324,848	Borges Völker et al. (2019)	↗
POS GSW (test)	7,320	113,565	Hollenstein and Aepli (2014)	↗
GDI (train)	14,279	112,707	Zampieri et al. (2019)	-
GDI (validation)	4,530	33,579	Zampieri et al. (2019)	-
GDI (test)	4,743	42,699	Zampieri et al. (2019)	-
Retrieval DE	1,997	50,833	Federmann et al. (2022)	↗
Retrieval GSW-BE	1,997	53,119	Aepli et al. (2023)	↗
Retrieval GSW-ZH	1,997	54,501	Aepli et al. (2023)	↗

Table A7: Dataset statistics for the downstream tasks.