hmBERT: Historical Multilingual Language Models for Named Entity Recognition

Stefan Schweter[1], Luisa März[2,3], Katharina Schmid[1] and Erion Çano[2]

[1]*Bayerische Staatsbibliothek München, Digital Library/ Munich Digitization Center, Munich, Germany*
[2]*Digital Philology, Research Group Data Mining and Machine Learning, University of Vienna, Austria*
[3]*Natural Language Processing Expert Center, Data:Lab, Volkswagen AG, Munich, Germany*

Abstract

Compared to standard Named Entity Recognition (NER), identifying persons, locations, and organizations in historical texts constitutes a big challenge. To obtain machine-readable corpora, the historical text is usually scanned and Optical Character Recognition (OCR) needs to be performed. As a result, the historical corpora contain errors. Also, entities like location or organization can change over time, which poses another challenge. Overall, historical texts come with several peculiarities that differ greatly from modern texts and large labeled corpora for training a neural tagger are hardly available for this domain. In this work, we tackle NER for historical German, English, French, Swedish, and Finnish by training large historical language models. We circumvent the need for large amounts of labeled data by using unlabeled data for pretraining a language model. We propose hmBert, a historical multilingual BERT-based language model, and release the model in several versions of different sizes. Furthermore, we evaluate the capability of hmBert by solving downstream NER as part of this year's HIPE-2022 shared task and provide detailed analysis and insights. For the Multilingual Classical Commentary coarse-grained NER challenge, our tagger *HISTeria* outperforms the other teams' models for two out of three languages.

Keywords

Named Entity Recognition, historical NER, Transformer-based language models, Historical texts, Flair

1. Introduction

Standard Named Entity Recognition (NER) for high resource domains has already been successfully addressed with performances above 90% F1-score [1, 2]. In contrast, NER taggers often fail to achieve satisfying results in the historical domain. Since historical datasets usually stem from Optical Character Recognition (OCR) and also include domain shifts, they contain characteristic errors not found in modern text. Low-resource fonts like Fraktur pose additional challenges for clean OCR. Another problem is that large amounts of labeled data are required when training neural models and only little labeled data exists for historical NER [3]. Because of all these challenges, systems designed for contemporary datasets cannot be applied to the historical domain without adaptations or further training. However, in the last few years, a number of works have shown that it is possible to adapt systems by using different approaches [4].

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
0000-0002-7190-2090 (S. Schweter); 0000-0003-4542-2437 (L. März); 0000-0001-6057-6640 (K. Schmid); 0000-0002-5496-3860 (E. Çano)

In this work, we develop a new BERT-based language model [5] for the historical context: hmBert. We tackle NER for historical German, English, French, Swedish, and Finnish. We use self-supervised learning to pretrain our language model on unlabeled data before we fine-tune the NER tagger on labeled data. This allows to reduce the need for large amounts of labeled training data. To mitigate the impact of OCR noise in the pretraining corpora, we use a filtering step that allows to control the OCR confidence of the texts.

Another design step in training language models is the choice of the underlying vocabulary. Beltagy et al. [6] showed that using a domain-specific vocabulary leads to performance improvements compared to using a general domain vocabulary. Thus, we use a sub-corpus of our pretraining corpus to create an in-domain vocabulary for the hmBert training. Finally, we arrive with a powerful hmBert model that establishes state-of-the-art results for three out of four languages on the NewsEye NER dataset [7]. As large language models require a lot of computational resources for pretraining and during inference time, we also provide smaller models.

Addressing the HIPE-2022 NERC-Coarse task, we also study a single-model vs. one-model approach. Our comparison shows that fine-tuning hmBert models for each language individually (single-model approach) improves performance compared to models that were fine-tuned on data from all languages (one-model approach). At the same time, however, fine-tuning individual models is much more computationally expensive. The one-model approach is more efficient, achieving similar performance while requiring less computation.

In addition, our final model *HISTeria* is trained by using multi-stage fine-tuning. We first fine-tune the multilingual model and evaluate it over the development data of all the different available languages. The resulting hyperparameter configuration is used for another fine-tuning step for each monolingual model. Finally, our detailed study of hmBert also includes experiments with a knowledge-based approach, training an ELECTRA-based language model [8], and addressing a tokenization issue. These additional experiments did not enhance performance but represent a suitable starting point for further research.

Our contributions are i) the comprehensive description of the development of hmBert, ii) the release of hmBert models of different sizes, iii) the release of the hmBert pretraining code, and iv) extensive experiments using hmBert including detailed insights for the community.

This paper is structured as follows: the next section (2) describes hmBert and its development in details. We include an explanation of the used datasets, as well as processing, hyperparameter settings, and pretraining steps. We close the section with a downstream task evaluation. Section 3 provides insights into the HIPE-2022 Multilingual Commentary Challenge[1] [9]. We describe our approach for the shared task submission in detail and provide an analysis of our results. We conclude the paper with Section 4.

2. hmBert: Historical Multilingual BERT Model

In this section, we present hmBert which supports German, English, French, Finnish, and Swedish. We train two different models with different vocabulary sizes: 32,000 and 64,000. We first describe the corpora used for training hmBert, as well as preprocessing and filtering

[1] https://hipe-eval.github.io/HIPE-2022/

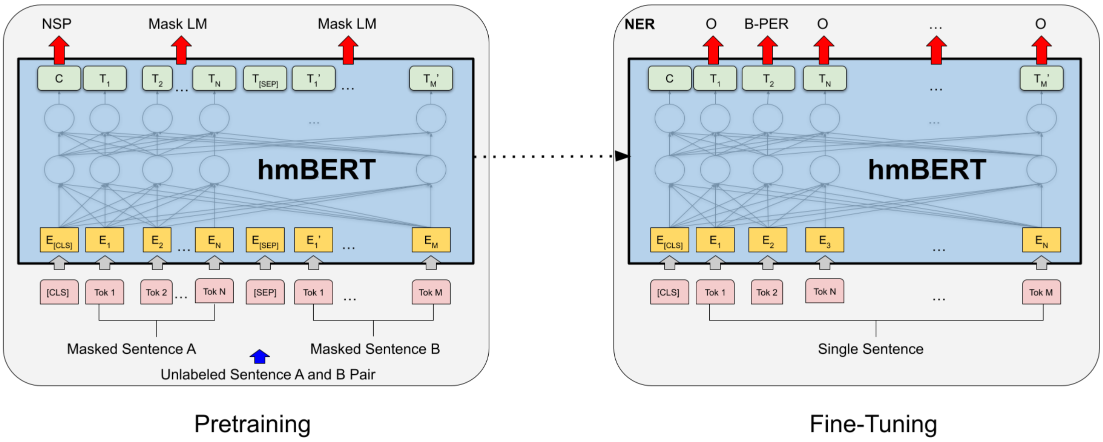

Figure 1: Overall pretraining of our $hmBERT_{32k}$ model and fine-tuning procedure for NER downstream tasks. CLS is a special symbol added in front of every input example, and SEP is a special separator token.

steps to create the pretraining corpus. In addition, we explain the pretraining process and end the section by evaluating the model on a downstream NER task. Figure 1 shows the overall pretraining procedure for hmBERT and its application on downstream tasks.

2.1. Corpora

For German, French, Swedish and Finnish we use the Europeana newspapers[2] provided by the European Library. For English we use a dataset published by the British Library [10]. The dataset contains OCR-processed text from digitized books and has also been used by Hosseini et al. [11] to train historical language models for English.

2.1.1. Filtering

OCR full-text for the Europeana newspapers also includes an OCR confidence value. This measure indicates the average OCR confidence for each word of a newspaper[3]. For German and French we perform a number of characters per year analysis using different (minimum required) OCR confidence thresholds. For German, we test three different thresholds and report the resulting dataset size (see Table 15 in the appendix). We use an OCR confidence threshold of 0.60 to get a final dataset of approx. 28 GB. For French, we test five different OCR confidence values (see Table 16 in the appendix) and choose 0.70 so that the resulting dataset size of 27 GB is comparable to the size of the German dataset. For Finnish and Swedish, we use an OCR confidence threshold of 0.60. However, training data for Swedish and Finnish is very limited. In total, only 1.2 GB for Finnish, and 1.1 GB for Swedish are available, thus these corpora are not filtered any further using other OCR confidence thresholds. For English, language filtering using

[2] http://www.europeana-newspapers.eu/
[3] https://www.clarin.eu/sites/default/files/Nuno_Freire_Europeana_CLARINPLUS.pdf

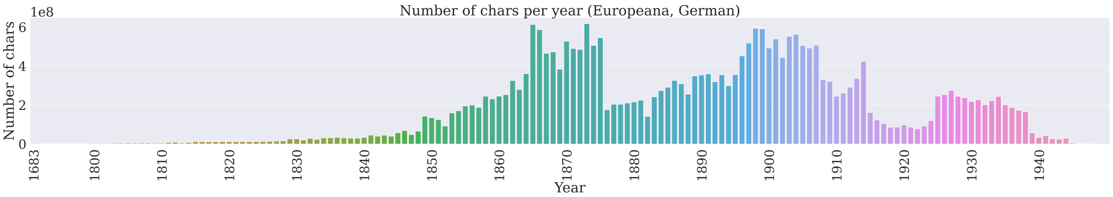

Figure 2: Number of characters per year distribution for filtered German Europeana corpus (1683-1949).

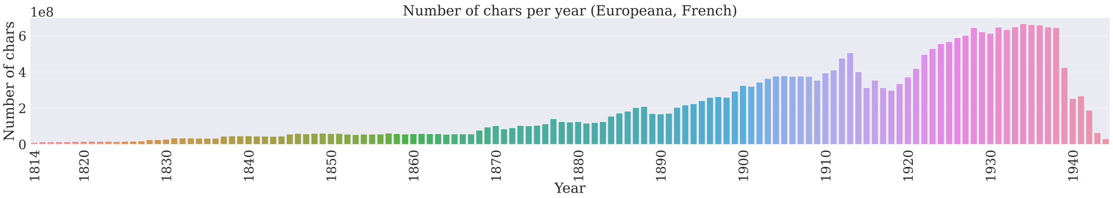

Figure 3: Number of characters per year distribution for filtered French Europeana corpus (1814-1944).

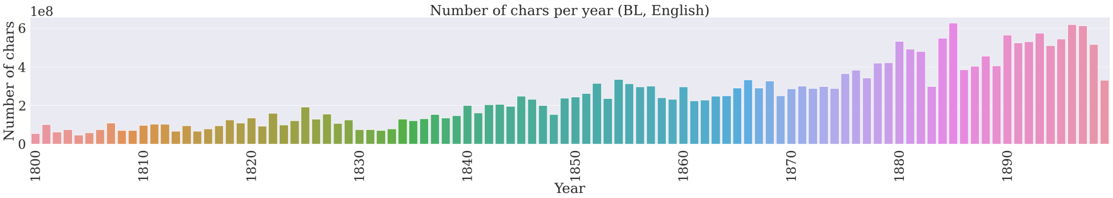

Figure 4: Number of characters per year distribution for filtered English corpus from British Library (1800-1899).

langdetect[4] for each book in the corpus is performed. Additionally, we use books published between 1800 and 1900 exclusively. The resulting English corpus has a total size of 24 GB.

To get a deeper insight into the filtered corpora, we analyze the distribution of characters over time for each language. Figure 2 shows the distribution for German. The period from 1865 to 1914 is well-covered in the dataset, while the years from 1683 to 1849 and the 20th century are underrepresented. For French, the 20th century is highly covered, but there is only little data available for the 19th century (see Figure 3), which contrasts with the German corpus. The English corpus contains texts from the 19th century only and shows good coverage starting from 1850. However, there is only little coverage from 1800 to 1849 (see Figure 4). Since both Finnish and Swedish corpora include newspapers from 1900 to 1910 only, we do not analyze the number of characters per year for these datasets.

2.1.2. Multilingual Vocabulary Generation

To create a BERT-compatible wordpiece-based vocabulary [12], we use 10GB of each language and train the vocabulary using the Hugging Face Tokenizers library[5]. We build a cased vocabu-

[4] https://github.com/Mimino666/langdetect
[5] https://github.com/huggingface/tokenizers

Table 1
NER datasets that are used for calculating subword fertility rate and portion of UNKs. For English, the development dataset was used due to a missing training split.

Language	NER Corpora
German	CLEF-HIPE-2020 [15], NewsEye [7]
French	CLEF-HIPE-2020 [15], NewsEye [7]
English	CLEF-HIPE-2020 [15]
Finnish	NewsEye [7]
Swedish	NewsEye [7]

Table 2
Subword fertility rate and portion of UNKs calculated on NER datasets using a 32k wordpiece-based vocabulary.

Language	Subword Fertility	UNK Portion
German	1.43	0.0004
French	1.25	0.0001
English	1.25	0.0
Finnish	1.69	0.0007
Swedish	1.43	0.0

lary with no lower casing or accent stripping being performed. For Finnish and Swedish we need to upsample[6] the corpus because both corpora have a size of 1 GB only.

We create a 32k and 64k vocabulary. Inspired by Rust et al. [13], we report the subword fertility rate (SFR) and the portion of unknown (UNK) tokens per language on various historical NER datasets (see Table 1). The SFR is defined as the average number of subwords a tokenizer produces per word [13]. It indicates how aggressively a tokenizer splits, i.e. whether it over-segments or not. As over-segmentation can negatively impact downstream performance, an SFR close to 1 (indicating that the tokenizer vocabulary contains every word in the input text) is optimal. UNK tokens are challenging because such tokens are not seen during pretraining and the model cannot provide useful information for them during the fine-tuning phase [14]. Table 2 and Table 3 show the SFR and portion of UNKs in the 32k/64k corpus. French and English have the lowest SFRs, whereas Finnish has the highest rate in both wordpiece-based vocabularies.

2.2. Final Pretraining Corpus

For common multilingual models such as multilingual BERT [mBERT; 5], XLM-RoBERTa [16] or mT5 [17] different corpus sampling strategies have been developed to up-/downsample low-/high-resource languages [18]. Since our multilingual language model includes five languages only (mBERT covers 104 languages[7]), we use a similar size for all languages. After upsampling the Swedish and Finnish corpora to 27GB each, we arrive at a total dataset size of 130 GB. Table

[6] For upsampling we simply concatenate the original corpus N-times to match the desired 10 GB size per language.

[7] https://github.com/google-research/bert/blob/master/multilingual.md

Table 3
Subword fertility rate and portion of UNKs calculated on NER datasets using a 64k wordpiece-based vocabulary.

Language	Subword Fertility	UNK Portion
German	1.31	0.0004
French	1.16	0.0001
English	1.17	0.0
Finnish	1.54	0.0007
Swedish	1.32	0.0

Table 4
Size per language of final pretraining corpus for hmBert.

Language	Dataset Size
German	28GB
French	27GB
English	24GB
Finnish	27GB
Swedish	27GB
Total	130GB

4 shows an overview of the sizes per language included in our final pretraining corpus. For the hmBert model with a vocabulary size of 32k, we use the official BERT implementation[8] to create pretraining data. Detailed description of all parameters used for the creation of pretraining data can be found in Section A.2 of the appendix.

2.3. Models

We pretrain an hmBert model with a vocabulary size of 32k, further denoted as $hmBERT_{32k}$, and another hmBert model with a vocabulary size of 64k, further denoted as $hmBERT_{64k}$. Inspired by Hou et al. [19], we also pretrain and release smaller hmBert models, with the number of layers ranging from 2 to 8 and hidden sizes ranging from 128 to 512. Pretraining of the different models is described in detail in Section A.3 of the appendix.

2.4. Downstream Task Evaluation

We evaluate the $hmBERT_{32k}$ models on the NewsEye NER dataset [7], because this dataset includes most of the languages that hmBert covers (except English), and compare them with the current state-of-the-art reported by Hamdi et al. [20]. We use the Flair [21] library and perform a hyperparameter search (see Table 18 in appendix) using the common fine-tuning paradigm. Fine-tuning adds a single linear layer to a Transformer and fine-tunes the entire architecture on the NER downstream task. To bridge the difference between subword modeling

[8] https://github.com/google-research/bert#pre-training-with-bert

Table 5
Performance overview of $hmBERT_{32k}$ models on German NewsEye NER dataset.

Model Name	Development F1-Score	Test F1-Score
$hmBERT_{32k}$ Tiny	30.16	24.35
$hmBERT_{32k}$ Mini	35.74	31.54
$hmBERT_{32k}$ Small	40.27	39.04
$hmBERT_{32k}$ Medium	43.45	43,41
$hmBERT_{32k}$ Base	46.17	46.66
Hamdi et al. [20]	-	**48.3**

and token-level predictions, subword pooling is applied to create token-level presentations which are then passed to the final linear layer. A common subword pooling strategy is to use the first subtoken to represent the entire token and we also use this strategy in our experiments. To train our architecture, we use AdamW [22] optimizer, a very small learning rate and a fixed number of epochs as a hard-stopping criterion. We evaluate the model performance after each training epoch on the development set and use the best model (strict micro F1-score) for final evaluation. We adopt a one-cycle [23] training strategy, in which the learning rate linearly decreases until it reaches 0 by the end of the training. Tables 5 - 8 show the performance of our $hmBERT_{32k}$ models compared to the current state-of-the-art.

For German, even the $hmBERT_{32k}$ base model could not reach the performance reported by Hamdi et al. [20], that was based on the models developed by Boros et al. [24]. The performance difference is 1.64 percentage points. This could be due to the fact that the German NewsEye dataset is very large and the hyperparameter search needed to be extended. Furthermore, Hamdi et al. [20] proposed a new architecture for handling OCR errors by adding two extra transformer layers, whereas we only performed a standard fine-tuning approach. For French our $hmBERT_{32k}$ medium sized model is very close to the result reported by Hamdi et al. [20]. The $hmBERT_{32k}$ base model outperforms the current best result by +2.7 percentage points. The same performance gain can be observed for Finnish and Swedish: The $hmBERT_{32k}$ base model outperforms the current SOTA by 2.41 percentage points for Finnish, and 2.1 percentage points for the Swedish NewsEye dataset. Figure 5 shows an overall performance comparison for the pretrained $hmBERT_{32k}$ smaller models on the NewsEye dataset. On average, the performance difference between the 8-layer $hmBERT_{32k}$ medium and the 12-layer $hmBERT_{32k}$ base model is 2.7 percentage points.

3. HIPE-2022: Multilingual Classical Commentary Challenge

We participated in the Multilingual Classical Commentary Challenge (MCC) that was newly introduced in the 2022 edition of HIPE [25] with our tagger being denoted as *HISTeria*. The challenge requires participants to work with historical classical commentaries in at least two different languages and to develop solutions for Named Entity Recognition, Classification, and/or Linking. *HISTeria* aims to detect and classify named entities according to coarse-grained types (NERC-Coarse task) and is described in more detail in this section.

Table 6
Performance overview of $hmBERT_{32k}$ models on French NewsEye NER dataset.

Model Name	Development F1-Score	Test F1-Score
$hmBERT_{32k}$ Tiny	60.04	50.79
$hmBERT_{32k}$ Mini	70.55	62.28
$hmBERT_{32k}$ Small	75.72	69.02
$hmBERT_{32k}$ Medium	78.99	72.51
$hmBERT_{32k}$ Base	81.58	**75.10**
Hamdi et al. [20]	-	72.7

Table 7
Performance overview of $hmBERT_{32k}$ models on Finnish NewsEye NER dataset.

Model Name	Development F1-Score	Test F1-Score
$hmBERT_{32k}$ Tiny	30.37	34.76
$hmBERT_{32k}$ Mini	56.60	62.68
$hmBERT_{32k}$ Small	64.31	73.20
$hmBERT_{32k}$ Medium	69.95	76.34
$hmBERT_{32k}$ Base	76.05	**80.11**
Hamdi et al. [20]	-	77.7

Table 8
Performance overview of $hmBERT_{32k}$ models on Swedish NewsEye NER dataset.

Model Name	Development F1-Score	Test F1-Score
$hmBERT_{32k}$ Tiny	43.65	38.91
$hmBERT_{32k}$ Mini	64.05	65.58
$hmBERT_{32k}$ Small	73.47	76.29
$hmBERT_{32k}$ Medium	78.07	82.47
$hmBERT_{32k}$ Base	81.13	**83.60**
Hamdi et al. [20]	-	81.5

3.1. Data

A classical commentary is a scholarly publication that aims to facilitate the reading and understanding of classical works of literature by providing additional information such as translations or bibliographic references. Apart from the challenges that are common to historical texts, commentaries have other characteristics that may complicate Named Entity Recognition and Classification: they frequently cite the original literary text, making them inherently multilingual, and they often use abbreviations to convey information more concisely. For the Multilingual Classical Commentary Challenge, HIPE[9] has chosen a single dataset that was

[9] https://github.com/hipe-eval/HIPE-2022-data/blob/main/documentation/README-ajmc.md

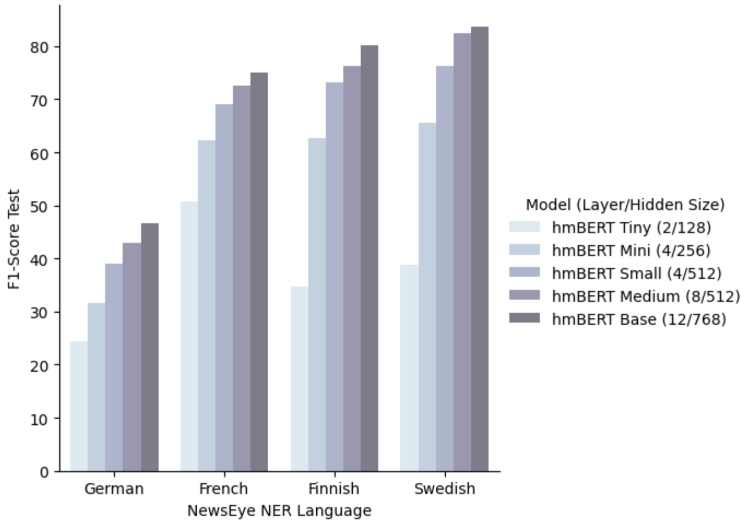

Figure 5: Overview of performance of $hmBERT_{32k}$ smaller models on NewsEye NER datasets. F1-score on the test set is reported here.

Table 9
Dataset statistics about `ajmc` dataset.

Language	Training Sentences	Development Sentences
German	1,024	192
English	1,154	252
French	894	202

created in the context of the Ajax MultiCommentary project[10] (`ajmc` dataset). The dataset contains excerpts from commentaries published in the 19th century in English, French, and German. The French texts date from 1886, the German ones from 1853 and 1894, and the English ones from 1881 and 1896. This emphasis on the second half of the 19th century fits well with the temporal distribution of our pretraining data for English and German. Apart from standard entity types like *person* or *location*, the dataset also includes domain-specific annotations like the *scope* and *work* entity type for bibliographic references. Additional dataset statistics can be found in Table 9 and in the HIPE 2022 Overview paper [9].

3.2. Single-Models vs. One-Model Approach

In preliminary experiments models that are independently fine-tuned for each language (single-model approach) and a model that uses training data from all languages (one-model approach) are compared. We perform hyperparameter searches for the two approaches. The relevant hyperparameters for fine-tuning models are shown in the appendix (Table 19). We use the Flair library for all experiments. For the one-model approach, a breakdown analysis for each language

[10] https://mromanello.github.io/ajax-multi-commentary/

Table 10
Performance comparison for NERC-coarse between single-model and one-model approach on `ajmc` development dataset. Numbers express F1-score calculated by using the strict evaluation regime.

Language	Single-Model	One-Model
German	86.21	**86.68**
English	**84.98**	84.85
French	**85.69**	85.09

is performed after determining the best hyperparameter configuration. This is compared to the three independently fine-tuned models for each language. For German, the one-model approach is +0.47 percentage points better than the single-model approach. For English, the one-model approach performs slightly worse (-0.13 percentage points) and for French, the single-model approach outperforms the one-model by 0.6 percentage points. However, the single-model approach requires fine-tuning of 120 models, whereas the one-model approach only needs 40 models to be fine-tuned for hyperparameter search. To save resources, we decided to use the one-model approach for further experiments. The performance comparison on the `ajmc` dataset is shown in Table 10.

3.3. Multi-Stage Fine-Tuning

Wang et al. [26] proposed a knowledge-based system for multilingual NER using a multi-stage fine-tuning approach for the MultiCoNER SemEval 2022 task[11]. The first stage of multi-stage fine-tuning refers to training a multilingual model on data from different languages. In the second stage, this fine-tuned multilingual model is used as a starting point for training a monolingual model. We adapt this approach for our final system: in the first stage, we fine-tune one multilingual model over the training data of all three languages (German, English, and French) and optimize over all development data (one-model approach) using a hyperparameter search. We select the best hyperparameter configuration as a combination of batch size, the number of epochs, and the learning rate, which results in five models (because of five different random seeds). The hyperparameter search grid for the different stages is shown in Section B.1 in the appendix. From these five models, we choose the one with the highest F1-score on the development set for second stage fine-tunings. In the second stage, we use the best model from the first stage and fine-tune single models for each language with a hyperparameter search on the development set. For each language, we select the best hyperparameter configuration and choose the best performing model with the highest F1-score on the development set. In preliminary experiments, this multi-stage fine-tuning approach boosts performance by 1.23 percentage points on average compared to results in the first stage.

For our final submission, $hmBERT_{32k}$ achieves the best results during the first-stage of fine-tuning with a batch size of 4, 10 fine-tuning epochs and a learning rate of $5e-05$. This results in an average F1-score of 86.89 on the (combined) development sets for `ajmc`. The best hyperparameter configuration for $hmBERT_{64k}$ can be achieved when using a batch size of 8,

[11] https://multiconer.github.io/

Table 11
Final results on `ajmc` development dataset for all languages using best models after multi-stage fine-tuning. Results are reported with official HIPE scorer.

Submission ID	Hyperparameter Configuration	Strict F1-Score	Fuzzy F1-Score
German ($hmBERT_{32k}$) - 1	bs8-e05-lr3e-05	91.5	**94.2**
German ($hmBERT_{64k}$) - 2	bs8-e10-lr3e-05	**92.0**	93.9
English ($hmBERT_{32k}$) - 1	bs4-e10-lr3e-05	**89.1**	92.9
English ($hmBERT_{64k}$) - 2	bs8-e10-lr3e-05	88.0	**93.8**
French ($hmBERT_{32k}$) - 1	bs4-e10-lr3e-05	**86.8**	**93.1**
French ($hmBERT_{64k}$) - 2	bs4-e10-lr5e-05	85.9	93.0

10 epochs of fine-tuning and a learning rate of $3e-05$. This results in an overall F1-score of 86.69 percentage points. Thus, $hmBERT_{64k}$ is slightly worse than $hmBERT_{32k}$ (-0.2 percentage points). Table 11 shows the performance for our final submissions using $hmBERT_{32k}$ and $hmBERT_{64k}$ for all languages in the `ajmc` dataset. We report strict and fuzzy F1-scores using the official HIPE-scorer[12]. We exclude document-level scores for better readability.

3.4. *HISTeria* Results

Table 12 shows an overview of *HISTeria* compared to the runs of other teams in the HIPE-2022 shared task[13] .

To gain a better understanding of our models, we use the attribute-aided evaluation proposed by Fu et al. [27]. In order to highlight the strengths and weaknesses of different models, they analyze how model performance varies with regard to certain attributes. In the case of NER, properties that may influence performance are i) how consistently a given surface form of a token or an entity is labelled across a dataset (`tCon` and `eCon`), ii) how often a given token or entity appears in the dataset (`tFre` and `eFre`), iii) the number of tokens that make up an entity (`eLen`) or sentence (`sLen`) as well as iv) the relative number of out-of-vocabulary words and entities per sentence (`oDen` and `eDen`). Using the implementation by Fu et al. [27], we distribute the values into buckets and compute the strict F1-score for each bucket. Table 13 shows Spearman's rank correlation coefficient as a measure of how well the attribute correlates with the F1-score, and the standard deviation of the F1-score to indicate how strongly the attribute influences performance. We omit results that are not statistically significant.

For the two German models, none of the attributes seem to correlate with performance in a statistically significant way. For the English and French models, performance correlates directly and positively with the consistency of the token labels. The standard deviation of 10% (French) and 8-9% (English) of the F1-score indicates that this attribute has a marked impact on performance. For French $hmBERT_{32k}$, entity length influences performance to the same degree. In this case, performance gets worse the more tokens an entity has. The impact of entity length on English $hmBERT_{32k}$ and $hmBERT_{64k}$ is less strong but still notable (standard

[12] https://github.com/hipe-eval/HIPE-scorer
[13] https://github.com/hipe-eval/HIPE-2022-eval/

Table 12
Final results on `ajmc` test dataset for all languages compared to other participants in the HIPE-2022 shared task. HISTeria denotes our system. Rank is ordered by strict F1-score.

Rank	Language	Submission ID	Strict F1-Score	Fuzzy F1-Score
1	German	L3i (team 2) - 2	93.4	95.2
2	German	HISTeria ($hmBERT_{32k}$) - 1	91.3	93.7
3	German	HISTeria ($hmBERT_{64k}$) - 2	91.2	94.5
4	German	L3i (team 2) - 1	90.8	93.4
5	German	Neural baseline	81.8	87.3
1	English	HISTeria ($hmBERT_{64k}$) - 2	85.4	91.0
2	English	L3i (team 2) - 1	85.0	89.4
3	English	L3i (team 2) - 2	84.1	88.4
4	English	HISTeria ($hmBERT_{32k}$) - 1	81.9	89.9
5	English	Neural baseline	73.6	82.8
1	French	HISTeria ($hmBERT_{64k}$) - 2	84.2	88.0
2	French	HISTeria ($hmBERT_{32k}$) - 1	83.3	88.8
3	French	L3i (team 2) - 2	82.6	87.2
4	French	L3i (team 2) - 1	79.8	86.0
5	French	Neural baseline	74.1	82.5

Table 13
Spearman's rank correlation coefficient and standard deviation of models' F1-score depending on different attribute values. We omit results that are not statistically significant.

Model	Attribute	Spearman	Standard Deviation
English $hmBERT_{32k}$	`tCon`	1.0	0.09
	`eLen`	-1.0	0.06
	`oDen`	-1.0	0.09
English $hmBERT_{64k}$	`tCon`	1.0	0.08
	`eLen`	-1.0	0.01
French $hmBERT_{32k}$	`tCon`	1.0	0.10
	`eLen`	-1.0	0.10
French $hmBERT_{64k}$	`tCon`	1.0	0.10

deviation of 6% and 1% respectively). In addition to entity length, the amount of words that did not feature in the training set also correlates negatively with the performance of English $hmBERT_{32k}$.

3.5. Challenges

We also experimented with the knowledge-based system for multilingual NER that was proposed by Wang et al. [26]. We used their implementation to enrich the original `ajmc` datasets with a knowledge base and implemented their context approach in the Flair library. More precisely, we used the FLERT approach [28] and utilized the knowledge-base enriched context as the left context for each training example. A left context size of 128 performs best in the experiments. However, the final result was slightly worse than using no context at all. This may be due to the fact that a contemporary, general-purpose knowledge base (Wikipedia) was used. A domain-specific knowledge base may yield better results. As the preliminary results were slightly worse than our main baseline, we did not conduct further experiments with this knowledge-based system.

We calculated the portion of UNKs in the German `ajmc` dataset and found that the portion rate of 16.3 % is unreasonably high. We discovered that the German `ajmc` dataset contains long-s characters, unlike the Europeana Newspaper corpora which were used to train a vocabulary. As a consequence, the hmBert tokenizer is not able to handle tokens that include long-s characters, resulting in UNKs. For our final system, we manually replaced all long-s characters with a normal s character to circumvent the UNK problem. In upcoming versions of our hmBert models, we will add this replacement step in the tokenizer configuration directly. Furthermore, we also trained an ELECTRA model [8] for 1M steps on the same pretraining corpus as the $hmBERT_{32k}$ model. We found that the downstream performance on NewsEye datasets was 1 to 3 percentage points worse than $hmBERT_{32k}$ and -0.28 percentage points worse on the `ajmc` dataset. We have therefore decided not to release the model yet.

3.6. Community Contributions

To foster research on language and NER models for the historical domain, we publicly release our pretrained and fine-tuned models on the Hugging Face Model Hub[14] under the `dbmdz` namespace[15]. We also publicly release all code that was used for fine-tuning models[16]. Table 14 shows an overview of released models for our HIPE-2022 submission, including the model identifier on the Hugging Face Model Hub. All models are released under a permissive MIT license. Additionally, we added dataset support for all HIPE-2022 NER datasets into Flair library[17].

4. Conclusion

We presented hmBert, a new multilingual BERT-based language model for historical data. hmBert is composed of German, French, English, Finnish, and Swedish unsupervised corpora of historical OCR-processed texts. The corpora have been filtered for OCR confidence as well as sampled so that each language contributes a similar amount of data to the model. The underlying vocabulary is also derived from each of the languages used for hmBert. In our temporal analysis of the pretraining corpora, we have found that data from the 18th and 19th century is unevenly distributed across the different languages. For future models, we are looking for additional datasets to balance this representation. We evaluated two hmBert models of different sizes with downstream Named Entity Recognition. For the NewsEye dataset hmBert

[14] https://huggingface.co/

[15] https://huggingface.co/dbmdz

[16] https://github.com/dbmdz/clef-hipe

[17] Added in Flair version 0.11: https://github.com/flairNLP/flair/releases/tag/v0.11

Table 14
Community contributions for our HIPE-2022 submission: Pretrained language models and fine-tuned NER models are publicly available on the Hugging Face Model Hub.

Model Description	Model Name
$hmBERT_{32k}$ Tiny Model	`dbmdz/bert-tiny-historic-multilingual-cased`
$hmBERT_{32k}$ Mini Model	`dbmdz/bert-mini-historic-multilingual-cased`
$hmBERT_{32k}$ Small Model	`dbmdz/bert-small-historic-multilingual-cased`
$hmBERT_{32k}$ Medium Model	`dbmdz/bert-medium-historic-multilingual-cased`
$hmBERT_{32k}$ Base Model	`dbmdz/bert-base-historic-multilingual-cased`
$hmBERT_{64k}$ Base Model	`dbmdz/bert-base-historic-multilingual-64k-td-cased`
NER First Stage ($hmBERT_{32k}$)	`dbmdz/flair-hipe-2022-ajmc-all`
NER First Stage ($hmBERT_{64k}$)	`dbmdz/flair-hipe-2022-ajmc-all-64k`
NER Second Stage - German ($hmBERT_{32k}$)	`dbmdz/flair-hipe-2022-ajmc-de`
NER Second Stage - English ($hmBERT_{32k}$)	`dbmdz/flair-hipe-2022-ajmc-en`
NER Second Stage - French ($hmBERT_{32k}$)	`dbmdz/flair-hipe-2022-ajmc-fr`
NER Second Stage - German ($hmBERT_{64k}$)	`dbmdz/flair-hipe-2022-ajmc-de-64k`
NER Second Stage - English ($hmBERT_{64k}$)	`dbmdz/flair-hipe-2022-ajmc-en-64k`
NER Second Stage - French ($hmBERT_{64k}$)	`dbmdz/flair-hipe-2022-ajmc-fr-64k`

established a new state-of-the-art for three out of four languages: French, Finnish, and Swedish. For the 2022 HIPE Multilingual Classical Commentary Challenge, our *HISTeria* system could outperform the other systems for two out of three languages. Using multi-stage fine-tuning together with the multilingual BERT-based model led the model to its optimal performance. Detailed analysis showed the benefits of all of hmBerts design choices, as well as interesting findings for future research. Our contributions include all of the trained hmBert models and our source code, which are made publicly available.

Acknowledgments

We would like to thank Google's TPU Research Cloud (TRC) program for giving us access to TPUs that were used for training our hmBert models. We would also like to thank Hugging Face for providing the ability to host and perform inferencing of our models on the Hugging Face Model Hub.

References

[1] A. Akbik, T. Bergmann, R. Vollgraf, Pooled Contextualized Embeddings for Named Entity Recognition, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 724–728. doi:10.18653/v1/N19-1078.

[2] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, K. Tu, Automated Concatenation of Embeddings for Structured Prediction, in: the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International

Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), Association for Computational Linguistics, 2021.

[3] M. Ehrmann, G. Colavizza, Y. Rochat, F. Kaplan, Diachronic Evaluation of NER Systems on Old Newspapers, Bochumer Linguistische Arbeitsberichte, Bochum, Germany, 2016, pp. 97–107. URL: http://infoscience.epfl.ch/record/221391.

[4] M. Ehrmann, A. Hamdi, E. Linhares Pontes, M. Romanello, A. Douvet, A Survey of Named Entity Recognition and Classification in Historical Documents, ACM Computing Surveys (2022 (to appear)). URL: https://arxiv.org/abs/2109.11406.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[6] I. Beltagy, K. Lo, A. Cohan, SciBERT: A Pretrained Language Model for Scientific Text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. doi:10.18653/v1/D19-1371.

[7] A. Hamdi, E. L. Pontes, E. Boros, T. T. H. Nguyen, G. Hackl, J. G. Moreno, A. Doucet, Multilingual Dataset for Named Entity Recognition, Entity Linking and Stance Detection in Historical Newspapers, 2021. doi:10.5281/zenodo.4573313.

[8] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, in: International Conference on Learning Representations, 2020. URL: https://openreview.net/forum?id=r1xMH1BtvB.

[9] M. Ehrmann, M. Romanello, S. Najem-Meyer, A. Doucet, S. Clematide, Overview of HIPE-2022: Named Entity Recognition and Linking in Multilingual Historical Documents, in: A. Barrón-Cedeño, G. Da San Martino, M. Degli Esposti, F. Sebastiani, C. Macdonald, G. Pasi, A. Hanbury, M. Potthast, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), Lecture Notes in Computer Science (LNCS), Springer, 2022.

[10] B. L. Labs, Digitised books. c. 1510 - c. 1946. json (ocr derived text)., 2016. doi:10.21250/DB14.

[11] K. Hosseini, K. Beelen, G. Colavizza, M. Coll Ardanuy, Neural Language Models for Nineteenth-Century English, arXiv e-prints (2021) arXiv:2105.11321. arXiv:2105.11321.

[12] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, CoRR abs/1609.08144 (2016). URL: http://arxiv.org/abs/1609.08144. arXiv:1609.08144.

[13] P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, I. Gurevych, How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models, in: 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), Association for Computational Linguistics, Online, 2021, pp. 3118–3135. doi:10.18653/v1/2021.acl-long.243.

[14] J. Pfeiffer, I. Vulić, I. Gurevych, S. Ruder, UNKs Everywhere: Adapting Multilingual Language Models to New Scripts, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 10186–10203. doi:10.18653/v1/2021.emnlp-main.800.

[15] M. Ehrmann, M. Romanello, A. Flückiger, S. Clematide, Extended Overview of CLEF HIPE 2020: Named Entity Processing on Historical Newspapers, volume 2696 of *CEUR Workshop Proceedings. 2696*, CEUR-WS, 2020, p. 38. doi:10.5281/zenodo.4117566.

[16] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. doi:10.18653/v1/2020.acl-main.747.

[17] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 483–498. doi:10.18653/v1/2021.naacl-main.41.

[18] A. Conneau, G. Lample, Cross-lingual Language Model Pretraining, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf.

[19] L. Hou, R. Yuanzhe Pang, T. Zhou, Y. Wu, X. Song, X. Song, D. Zhou, Token Dropping for Efficient BERT Pretraining, arXiv e-prints (2022) arXiv:2203.13240. arXiv:2203.13240.

[20] A. Hamdi, E. Boroş, E. L. Pontes, T. T. H. Nguyen, G. Hackl, J. G. Moreno, A. Doucet, A Multilingual Dataset for Named Entity Recognition, Entity Linking and Stance Detection in Historical Newspapers, in: Proceedings of the 44rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021.

[21] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf, FLAIR: An easy-to-use framework for state-of-the-art NLP, in: NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), 2019, pp. 54–59.

[22] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[23] L. N. Smith, A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, arXiv e-prints (2018) arXiv:1803.09820. arXiv:1803.09820.

[24] E. Boros, A. Hamdi, E. Linhares Pontes, L. A. Cabrera-Diego, J. G. Moreno, N. Sidere, A. Doucet, Alleviating Digitization Errors in Named Entity Recognition for Historical Documents, in: Proc. of the 24th Conference on Computational Natural Language Learning,

ACL, 2020, pp. 431–441. URL: https://www.aclweb.org/anthology/2020.conll-1.35.

[25] M. Ehrmann, M. Romanello, S. Clematide, A. Doucet, Introducing the HIPE 2022 Shared Task:Named Entity Recognition and Linking in Multilingual Historical Documents, in: Proceedings of the 44^{d} European Conference on IR Research (ECIR 2022), Lecture Notes in Computer Science, Springer, Stavanger, Norway, 2022. URL: https://link.springer.com/chapter/10.1007/978-3-030-99739-7_44.

[26] X. Wang, Y. Shen, J. Cai, T. Wang, X. Wang, P. Xie, F. Huang, W. Lu, Y. Zhuang, K. Tu, W. Lu, Y. Jiang, DAMO-NLP at SemEval-2022 Task 11: A Knowledge-based System for Multilingual Named Entity Recognition (2022). URL: https://arxiv.org/abs/2112.06482. arXiv:2203.00545.

[27] J. Fu, P. Liu, G. Neubig, Interpretable Multi-dataset Evaluation for Named Entity Recognition, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6058–6069. doi:10.18653/v1/2020.emnlp-main.489.

[28] S. Schweter, A. Akbik, FLERT: Document-Level Features for Named Entity Recognition, 2020. arXiv:2011.06993.

[29] S. Schweter, BERTurk - BERT models for Turkish, 2020. doi:10.5281/zenodo.3770924.

[30] L. Hou, R. Y. Pang, T. Zhou, Y. Wu, X. Song, X. Song, D. Zhou, Token Dropping for Efficient BERT Pretraining, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3774–3784. URL: https://aclanthology.org/2022.acl-long.262.

A. hmBERT: Historical Multilingual BERT Model

A.1. Corpora Filtering

Table 15
Word level OCR confidence thresholds for German. Bold OCR confidence is used for the final corpus.

OCR Confidence	Dataset Size
0.60	28GB
0.65	18GB
0.70	13GB

Table 16
Word level OCR confidence thresholds for French. Bold OCR confidence is used for the final corpus.

OCR Confidence	Dataset Size
0.60	31GB
0.65	27GB
0.70	27GB
0.75	23GB
0.80	11GB

A.2. Final Pretraining Corpus

For creation of the pretraining data, we use the same parameters as BERTurk [29]: maximum sequence length = 512, maximum predictions per sequence = 75, masked language probability rate = 0.15, duplication factor = 5. Due to hardware limitations, we split the pretraining corpus into chunks of 1GB and create pretraining data for each chunk individually. For the hmBert model with a vocabulary size of 64k we use the official implementation[18] with the same parameters as for the 32k model, but we increase the maximum predictions per sequence to 76.

A.3. Models

We use the official BERT implementation[19] for pretraining $hmBERT_{32k}$. $hmBERT_{64k}$ is trained with the recently proposed "token dropping" approach by Hou et al. [30]. Using this approach, unimportant tokens starting from an intermediate layer in the model are dropped to make the model focus on important tokens more efficiently, which makes model pretraining faster compared to the original BERT implementation. For both pretraining approaches, we use a maximum sequence length of 512 for the full training time. For the pretraining of $hmBERT_{32k}$ a batch size of 128 is used for 3M training steps. Pretraining was done on a v3-32 TPU pod

[18] https://github.com/tensorflow/models/blob/27fb855b027ead16d2616dcb59c67409a2176b7f/official/legacy/bert/README.md#pre-training

[19] https://github.com/google-research/bert

within 67 hours. The pretraining of $hmBERT_{64k}$ was done on a single v4-8 TPU with a batch size of 512 for 1M steps within 114 hours. Figure 6 shows the pretraining loss for $hmBERT_{32k}$ and $hmBERT_{64k}$. The final $hmBERT_{32k}$ has 110.62M, whereas $hmBERT_{64k}$ has 135.19M parameters due to the increased vocabulary size.

For better comparability, we measure the number of total subtokens seen during pretraining[20] and the number of total subtokens of the pretraining corpus for our two hmBert models. More precisely, $hmBERT_{32k}$ has seen 196B subtokens during pretraining, whereas the pretraining corpus has a total size of 42B subtokens. This results in 4.7 pretraining epochs over the corpus. Our $hmBERT_{64k}$ model has seen 262B subtokens during pretraining. Because of the larger vocabulary size, the number of subtokens for the corpus is 39B. This results in 6.7 pretraining epochs over the corpus.

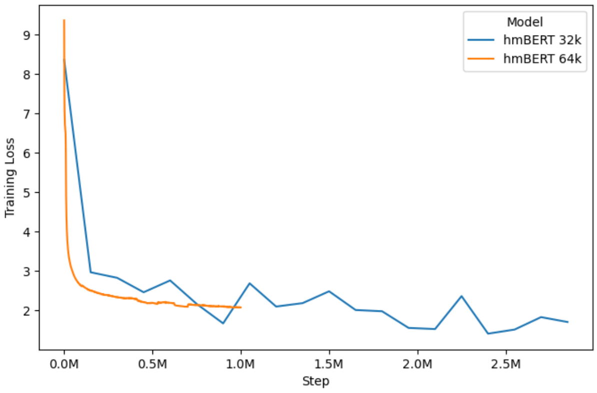

Figure 6: Overview of pretraining loss for $hmBERT_{32k}$ and $hmBERT_{64k}$.

For the smaller models, we use the same pretraining data and hyperparameter as for the base $hmBERT_{32k}$ model and pretrain them on a v3-32 TPU pod. Table 17 shows an overview of pretrained models, including their model size, number of parameters and pretraining time. Figure 7 shows an overview of pretraining loss for all smaller $hmBERT_{32k}$ models.

Table 17
Overview of smaller $hmBERT_{32k}$ models with their corresponding model size, number of parameters and pretraining time.

Model Name	Number of Layers	Hidden Size	Parameters	Pretraining Time
$hmBERT_{32k}$ Tiny	2	128	4.58M	4.3s / 1k steps
$hmBERT_{32k}$ Mini	4	256	11.55M	10.5s / 1k steps
$hmBERT_{32k}$ Small	4	512	29.52M	20.7s / 1k steps
$hmBERT_{32k}$ Medium	8	512	42.13M	35.0s / 1k steps
$hmBERT_{32k}$ Base	12	768	110.62M	80.0s / 1k steps

[20] Total number of subtokens during pretraining can be calculated as multiplication of training steps, batch size and sequence length

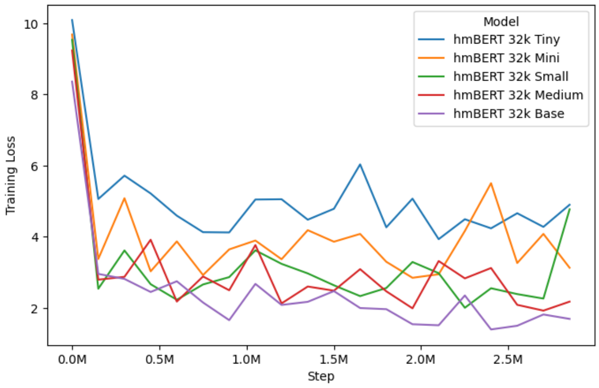

Figure 7: Overview of pretraining loss for smaller $hmBERT_{32k}$ models.

A.4. Downstream Task Evaluation

Table 18
Hyperparameter search for downstream evaluation on NewsEye NER dataset.

Parameter	Values
Batch Size	$[4, 8]$
Epoch	$[5, 10]$
Learning Rate	$[3e-05, 5e-05]$
Seed	$[1, 2, 4, 5]$

B. HIPE-2022: Multilingual Classical Commentary Challenge

B.1. Multi-Stage Fine-Tuning

Table 19
Hyperparameter search during the first stage of NER model fine-tuning.

Parameter	Values
Batch Size	$[4, 8, 16]$
Epoch	$[10]$
Learning Rate	$[1e-05, 2e-05, 3e-05, 4e-05, 5e-05]$
Seed	$[1, 2, 4, 5]$

Table 20
Hyperparameter search during the second stage of NER model fine-tuning.

Parameter	Values
Batch Size	$[4, 8]$
Epoch	$[5, 10]$
Learning Rate	$[3e-05, 5e-05]$
Seed	$[1, 2, 4, 5]$

As a batch size of 16 and learning rates of $1e-05$ and $2e-05$ do not perform well, we exclude them when performing hyperparameter search with $hmBERT_{64k}$.