EstBERT: A Pretrained Language-Specific BERT for Estonian

Hasan Tanvir and **Claudia Kittask** and **Sandra Eiche** and **Kairit Sirts**
Institute of Computer Science
University of Tartu
Tartu, Estonia
hasantanvir79@gmail.com, {claudia.kittask,sandra.eiche,sirts}@ut.ee

Abstract

This paper presents EstBERT, a large pretrained transformer-based language-specific BERT model for Estonian. Recent work has evaluated multilingual BERT models on Estonian tasks and found them to outperform the baselines. Still, based on existing studies on other languages, a language-specific BERT model is expected to improve over the multilingual ones. We first describe the EstBERT pretraining process and then present the models' results based on the finetuned EstBERT for multiple NLP tasks, including POS and morphological tagging, dependency parsing, named entity recognition and text classification. The evaluation results show that the models based on EstBERT outperform multilingual BERT models on five tasks out of seven, providing further evidence towards a view that training language-specific BERT models are still useful, even when multilingual models are available.[1]

1 Introduction

Pretrained language models, such as BERT (Devlin et al., 2019) or ELMo (Peters et al., 2018), have become the essential building block for many NLP systems. These models are trained on large amounts of unannotated textual data, enabling them to capture the general regularities in the language and thus can be used as a basis for training the subsequent models for more specific NLP tasks. Bootstrapping NLP systems with pretraining is particularly relevant and holds the greatest promise for improvements in the setting of limited resources, either when working with tasks of limited annotated training data or less-resourced languages like Estonian.

Since the first publication and release of the large pretrained language models on English, considerable effort has been made to develop support for other languages. In this regard, multilingual BERT models, simultaneously trained on the text of many different languages, have been published, several of which also include the Estonian language (Devlin et al., 2019; Conneau et al., 2019; Sanh et al., 2019; Conneau and Lample, 2019). These multilingual models' performance was recently evaluated on several Estonian NLP tasks, including POS and morphological tagging, named entity recognition, and text classification (Kittask et al., 2020). The overall conclusions drawn from these experiments are in line with previously reported results on other languages, i.e., for many or even most tasks, multilingual BERT models help improve performance over baselines that do not use language model pretraining.

Besides multilingual models, language-specific BERT models have been trained for an increasing number of languages, including for instance Camem-BERT (Martin et al., 2020) and FlauBERT (Le et al., 2020) for French, FinBERT for Finnish (Virtanen et al., 2019), RobBERT (Delobelle et al., 2020) and BERTJe (de Vries et al., 2019) for Dutch, Chinese BERT (Cui et al., 2019), BETO for Spanish (Cañete et al., 2020), RuBERT for Russian (Kuratov and Arkhipov, 2019) and others. For a recent overview about these efforts refer to Nozza et al. (2020). Aggregating the results over different language-specific models and comparing them to those obtained with multilingual models shows that depending on the task, the average improvement of the language-specific BERT over the multilingual BERT varies from 0.70 accuracy points in paraphrase identification up to 6.37 in sentiment classification (Nozza et al., 2020). The overall conclusion one can draw from these results is that while existing multilingual BERT models can bring along improvements over language-specific baselines, using language-specific BERT models can further considerably improve the performance of various NLP tasks.

Following the line of reasoning presented above, we set forth to train EstBERT, a language-specific BERT model for Estonian. In the following sections, we first give details about the data used for BERT pretraining and then describe the pretraining process. Finally, we will provide evaluation results on the same tasks as presented by Kittask et al. (2020) on multilingual BERT models, which include POS and morphological tagging, named entity recognition and text classification. Additionally, we also train a dependency parser based on the spaCy system. Compared to multilingual

---

[1]The model is available via HuggingFace Transformers library: https://huggingface.co/tartuNLP/EstBERT

models, the EstBERT model achieves better results on five tasks out of seven, providing further evidence for the usefulness of pretraining language-specific BERT models. Additionally, we also compare with the Estonian WikiBERT, a recently published Estonian-specific BERT model trained on a relatively small Wikipedia data (Pyysalo et al., 2020). Compared to the Estonian WikiBERT model, the EstBERT achieves better results on six tasks out of seven, demonstrating the positive effect of the amount of pretraining data on the generalisability of the model.

2 Data Preparation

The first step for training the EstBERT model involves preparing a suitable unlabeled text corpus. This section describes both the steps we took to clean and filter the data and the process of generating the vocabulary and the pretraining examples.

2.1 Data Preprocessing

For training the EstBERT model, we used the Estonian National Corpus 2017 (Kallas and Koppel, 2018),[2] which was the largest Estonian language corpus available at the time. It consists of four sub-corpora: the Estonian Reference Corpus 1990-2008, the Estonian Web Corpus 2013, the Estonian Web Corpus 2017, and the Estonian Wikipedia Corpus 2017. The Estonian Reference corpus (ca 242M words) consists of a selection of electronic textual material, about 75% of the corpus contains newspaper texts, the rest 25% contains fiction, science and legislation texts. The Estonian Web Corpora 2013 and 2017 make up the largest part of the material and they contain texts collected from the Internet. The Estonian Wikipedia Corpus 2017 is the Estonian Wikipedia dump downloaded in 2017 and contains roughly 38M words. The top row of the Table 1 shows the initial statistics of the corpus.

We applied different cleaning and filtering techniques to preprocess the data. First, we used the corpus processing methods from EstNLTK (Laur et al., 2020), which is an open-source tool for Estonian natural language processing. Using the EstNLTK, all XML/HTML tags were removed from the text, also all documents with a language tag other than Estonian were removed. Additional non-Estonian documents were further filtered out using the language-detection library.[3] Next, all duplicate documents were removed. For that, we used hashing—all documents were lowercased, and then the hashed value of each document was subsequently stored into a set. Only those documents whose hash value did not yet exist in the set (i.e., the first document with each hash value) were retained. We also used the hand-written heuristics,[4] developed for preprocessing the data for training the FinBert model

	Documents	Sentences	Words
Initial	3.9M	87.6M	1340M
After cleanup	3.3M	75.7M	1154M

Table 1: Statistics of the corpus before and after the cleanup.

(Virtanen et al., 2019), to filter out documents with too few words, too many stopwords or punctuation marks, for instance. We applied the same thresholds as were used for Finnish BERT. Finally, the corpus was truecased by lemmatizing a copy of the corpus with EstNLTK tools and using the lemma's casing information to decide whether the word in the original corpus should be upper- or lowercase. The statistics of the corpus after the preprocessing and cleaning steps are in the bottom row of Table 1.

2.2 Vocabulary and Pretraining Example Generation

Originally, BERT uses the WordPiece tokeniser, which is not available open-source. Instead, we used the BPE tokeniser available in the open-source sentencepiece[5] library, which is the closest to the WordPiece algorithm, to construct a vocabulary of 50K subword tokens. Then, we used BERT tools[6] to create the pretraining examples for the BERT model in the TFRecord format. In order to enable parallel training on four GPUs, the data was split into four shards. Separate pretraining examples with sequences of length 128 and 512 were created, masking 15% of the input words in both cases. Thus, 20 and 77 words in maximum were masked in sequences of both lengths, respectively.

3 Evaluation Tasks

Before describing the EstBERT model pretraining process itself, we first describe the tasks used to both validate and evaluate our model. These tasks include the POS and morphological tagging, named entity recognition, and text classification. In the following subsection, we describe the available Estonian datasets for these tasks.

3.1 Part of Speech and Morphological Tagging

For part of speech (POS) and morphological tagging, we use the Estonian EDT treebank from the Universal Dependencies (UD) collection that contains annotations of lemmas, parts of speech, universal morphological features, dependency heads, and universal dependency labels. We use the UD version 2.5 to enable comparison with the experimental results of the multilingual BERT models reported by Kittask et al. (2020). We train models to predict both universal POS (UPOS) and language-specific POS (XPOS) tags as well as

morphological features. The pre-defined train/dev/test splits are used for training and evaluation. Table 2 shows the statistics of the treebank splits. The accuracy of the POS and morphological tagging tasks is evaluated with the `conll18_ud_eval` script from the CoNLL 2018 Shared Task.

	Train	Dev	Test
Sentences	31012	3128	6348
Tokens	344646	42722	48491

Table 2: Statistics of the UDv2.5 Estonian treebank.

3.2 Named Entity Recognition

Estonian named entity recognition (NER) corpus (Tkachenko et al., 2013) annotations cover three types of named entities: locations, organizations, and persons. It contains 572 news stories published in local online newspapers Postimees and Delfi, covering local and international news on various topics. Table 3 displays statistics of the training, development and test splits. The performance of the NER models is evaluated with the `conlleval` script from the CoNLL 2000 shared task.

	Tokens	PER	LOC	ORG	Total
Train	155981	6174	4749	4784	15707
Dev	32890	1115	918	742	2775
Test	28370	1201	644	619	2464

Table 3: Statistics of the Estonian NER corpus.

3.3 Sentiment and Rubric Classification

Estonian Valence corpus (Pajupuu et al., 2016) consists of 4085 news extracts from Postimees Daily. All documents in the corpus are labeled with both sentiment and rubric classes. There are nine rubrics: Opinion, Estonia, Life, Comments-Life, Comments-Estonia, Crime, Culture, Sports, and Abroad. The four sentiment labels include Positive, Negative, Neutral, and Ambiguous. We split the data into 70/10/20 training, development and test sets, stratified over both rubric and sentiment analysis. Table 4 and Table 5 show the statistics about the sentiment and rubric view of the classification dataset respectively.

4 Pretraining EstBERT

The EstBERT model was pretrained on the architecture identical to the BERT$_{Base}$ model with 12 transformer blocks with 768 hidden units each and 110M trainable parameters. It was pretrained on the Masked Language Modeling (MLM) and the Next Sentence Prediction (NSP) tasks as described by Devlin et al. (2019). In MLM, the probability of correctly predicting the randomly masked tokens is maximised. Because in the

	Train	Dev	Test	Total
Positive	612	87	175	874
Negative	1347	191	385	1923
Neutral	505	74	142	721
Ambiguous	385	55	110	550
Total	2849	407	812	4068

Table 4: Sentiment label statistics of the Estonian Valence corpus.

	Train	Dev	Test	Total
Opinion	676	96	192	964
Estonia	289	41	83	413
Life	364	52	101	517
Comments-Life	354	50	102	506
Comments-Estonia	351	50	100	501
Crime	146	21	42	209
Culture	182	27	51	260
Sports	269	39	77	385
Abroad	218	31	64	313
Total	2849	407	812	4068

Table 5: Rubric label statistics of the Estonian Valence corpus.

transformer architecture, the model can simultaneously see both the left and the right context of a masked word, optimizing the MLM gives the model a bidirectional understanding of a sentence, as opposed to only the left or right context provided by recurrent language models. The NSP involves optimizing a binary classification task to predict whether the two sequences in the input follow each other in the original text or not, where half of the time, the second sequence is the correct next sentence and the other half of the time the two sequences are unrelated. The models were trained on four NVIDIA Tesla V100 GPUs across two nodes of the High-performance Computing Center at the University of Tartu (University of Tartu, 2018).

The model was first trained with the sequence length of 128. Then we evaluated the checkpoints generated during pretraining on the tasks described in section 3 to choose the final model with that sequence length. Finally, the chosen model was used as a starting point for training the longer model with 512 sequence length. Thus, as a result of pretraining, two EstBERT models, one with maximum sequence length 128 and the other with maximum sequence length 512, were obtained. The following subsections describe these three steps in more detail.

4.1 Pretraining with Sequence Length 128

The model with the sequence length of 128 was pretrained for two phases, both for 900K steps. Although the number of training steps was chosen following Vir-
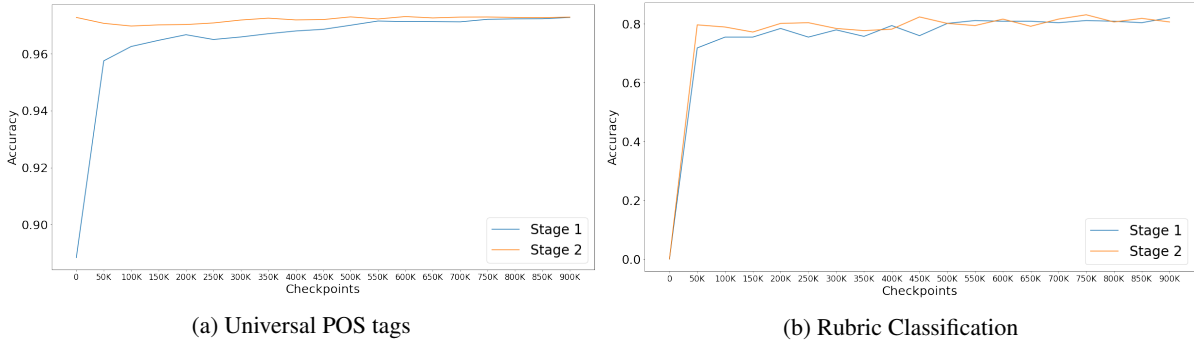
(a) Universal POS tags

(b) Rubric Classification

Figure 1: The validation performance on POS tagging and text classification tasks after every 50K checkpoints.

train_batch_size	32
max_seq_length	128
max_predictions_per_seq	20
num_train_steps	900000
num_warmup_steps	9000
learning_rate	1e-4
save_checkpoints_step	50000

Table 6: Hyperparameters used in the first two pretraining phases with the sequence length 128.

tanen et al. (2019), coincidentally this (900K steps) was the maximum number of steps we could fit in the given GPU time limit of 8 days. Therefore, the model was trained in two phases, each having 900K steps. A checkpoint was saved to the disk after every 50K steps. While the first phase of pretraining started from scratch with randomly initialised parameters, the second phase of training was initialised from the first phase's last checkpoint. Since the GPU memory availability was a major issue, the batch size was kept at 32 to avoid the tensors going beyond the allowed GPU memory size. The BERT_Base uses Adam optimiser with weight decay. For EstBERT, the same optimiser was used with warmup over the first 1% of steps (9000) to a peak learning rate of 1e-4. The relevant hyperparameters are shown in Table 6. The pretraining process took around 192 hours for each phase.

4.2 Pretraining Validation

During pretraining, a checkpoint was saved after every 50K steps for later evaluation. To monitor the pretraining process, we evaluated the performance of MLM, NSP, and the evaluation tasks described in section 3 on all these checkpoints.

For POS and morphological tagging, and named entity recognition, we finetuned EstBERT using scripts from HuggingFace transformers library.[7] A single randomly initialised fully connected classifier layer was trained on top of the token representations of the last hidden layer of the EstBERT model. All hyperparam-

eters were kept at their default values, which involves training for three epochs, using the learning rate of 5e-5 and batch size of 8. For the rubric and sentiment classification tasks, we adapted the classifier training scripts available in google's BERT repository.[8] The input to the single fully-connected classifier layer is the last hidden representation of the first token [CLS] in the input sequence. Here again, the classifier layer was initialised randomly and the default values for hyperparameters were used: training for three epochs with the learning rate 5e-5 and batch size 32. In all tasks, both the task-specific classification layer as well as the EstBERT parameters were finetuned.

The validation results of the masked language model, next sentence prediction accuracy, and all the evaluation tasks for all the eighteen checkpoints from stage one and other eighteen models from stage two were compared to pick the best model. The examples of validation curves for the UPOS tagging and the rubric classification tasks are shown in Figure 1. Although the checkpoint validation results from both phases showed more or less steady improvement with the increase of the number of steps trained, we observed that the checkpoint at 750K steps from phase two performs slightly better on all tasks than the rest of the checkpoints. Thus, this checkpoint was chosen as a final model with sequence length 128.

train_batch_size	16
max_seq_length	512
max_predictions_per_seq	77
num_train_steps	600000
num_warmup_steps	6000
learning_rate	1e-4
save_checkpoints_step	50000

Table 7: Hyperparameters used to pretrain the Est-BERT model with the sequence length 512.

---

[7]https://github.com/huggingface/transformers/blob/master/examples/token-classification/run_ner.py

[8]https://github.com/google-research/bert/blob/master/run_classifier.py/

Model	UPOS	XPOS	Morph	UPOS	XPOS	Morph
		Seq = 128			Seq = 512	
EstBERT	**<u>97.89</u>**	**98.40**	**96.93**	97.84	**<u>98.43</u>**	**<u>96.80</u>**
WikiBERT-et	97.78	98.36	96.71	97.76	98.35	96.67
mBERT	97.42	98.06	96.24	97.43	98.13	96.13
XLM-RoBERTa	97.78	98.36	96.53	97.80	98.40	96.69

Table 8: POS and morphological tagging accuracy on the Estonian UD test set. The highest scores in each column are in **bold**. The highest overall score of each task is <u>underlined</u>.

4.3 Pretraining with Sequence Length 512

The starting point for training the model with a sequence length of 512 was the final model chosen for the sequence length 128. The longer model was trained further up to 600K steps. The batch size was reduced to 16 as the size of the tensors would be larger for the sequence length 512 compared to 128. The hyperparameters used to train the longer model are shown in Table 7. During training, checkpoints were again saved after every 50K steps, and these were evaluated on all evaluation tasks as previously explained in Section 4.2. Based on these evaluations, the last checkpoint obtained after the 600K steps was chosen as the final model with 512 sequence length.

5 Results

The next subsections present the results obtained with the final EstBERT models with both sequence lengths on the tasks described in section 3. We follow the same setup of Kittask et al. (2020) to enable direct comparison with the multilingual models. Some additional steps were taken to prepare the Estonian Valence corpus. First, all duplicate items, 17 in total, were removed. Also, all items with the Ambiguous label were removed as retaining them has been shown to lower the the classification accuracy (Pajupuu et al., 2016). The same preprocessing was also applied in evaluating the multilingual BERT models for Estonian (Kittask et al., 2020).

For finetuning, we used the same scripts from the HuggingFace transformers repository that were used for the pretraining validation in section 4.2. The same scripts were also used to evaluate the multilingual models by Kittask et al. (2020). For each task, the learning rate of the AdamW optimiser and the batch size was tuned on the development set. The learning rate grid values were (5e-5, 3e-5, 1e-5, 5e-6, 3e-6) and the batch size grid values were (8, 16). The best model was found on the development set by using early stopping with the patience of 10 epochs.

We compare the results of EstBERT with the multilingual BERT models' results from Kittask et al. (2020) and the WikiBERT model trained on the Estonian Wikipedia (Pyysalo et al., 2020). WikiBERT-et model was finetuned using the same setup described above.

Model	Rubr.	Sent.	Rubr.	Sent.
		Seq = 128		Seq = 512
EstBERT	**<u>81.70</u>**	74.36	**80.96**	74.50
WikiBERT-et	72.72	68.09	71.13	69.37
mBERT	75.67	70.23	74.94	69.52
XLM-RoBERTa	80.34	**74.50**	78.62	**<u>76.07</u>**

Table 9: Rubric (Rubr.) and sentiment (Sent.) classification accuracy. The highest scores in each column are in **bold**. The highest overall score of each task is <u>underlined</u>.

5.1 POS and Morphological Tagging

The POS and morphological tagging results are summarised in Table 8 that shows the accuracy for universal POS tags (UPOS), language-specific POS tags (XPOS), and morphological features. The language-specific EstBERT outperforms all other models although the difference with the XLM-RoBERTa—the best-performing multilingual model—and the WikiBERT-et are quite small.

Similar to multilingual results, using longer sequence length on this task with the EstBERT model does not seem beneficial as the accuracy slightly increases only for XPOS tags but not for others. Overall, as the performances on these tasks are already very high, the absolute performance gains cannot be large. EstBERT obtains consistent improvements over mBERT, with the relative error reduction with both models on all tasks falling between 16-18%. The relative error reduction of the EstBERT compared to XLM-RoBERTa is smaller, in the range of 2-5%. The highest reduction of error of EstBERT compared to XLM-RoBERTa can be observed on the morphological tagging task with the shorter model where the relative error reduction is 12%. The WikiBERT-et model achieves almost identical results to XLM-RoBERTa with both sequence lengths.

5.2 Rubric and Sentiment Classification

The rubric and sentiment classification results are shown in Table 9. EstBERT outperforms mBERT and WikiBERT-et on both tasks by a large margin, but XLM-RoBERTa exceeds EstBERT on sentiment clas-

Model	Precicion	Recall	F1-Score	Precision	Recall	F1-Score
		Seq = 128			Seq = 512	
EstBERT	89.10	91.15	90.11	88.35	89.74	89.04
WikiBERT-et	**_89.86_**	90.83	**_90.34_**	88.31	**90.96**	**89.61**
mBERT	85.88	87.09	86.51	**88.47**	88.28	88.37
XLM-RoBERTa	87.55	**_91.19_**	89.34	87.50	90.76	89.10

Table 10: NER tagging results. Upper section shows the comparison between different models. The highest scores in each column are in **bold**. The highest overall score of each measure is underlined.

Entity	EstBERT			XLM-RoBERTa			WikiBERT-et		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
PER	94.80	**95.77**	95.28	**96.42**	94.45	**95.43**	94.87	94.45	94.66
ORG	78.38	**82.64**	80.45	75.48	82.12	78.66	**82.25**	81.61	**81.92**
LOC	**89.94**	91.38	90.66	86.06	**93.99**	89.85	88.89	92.99	**90.89**

Table 11: The entity-based scores for the EstBERT, XLM-RoBERTa and the WikiBERT-et models. The best scores are in **bold**.

sification. The difference between the two accuracy scores is relatively small when the model with sequence length 128 is used, but it increases when the longer sequence length is used.

Like XLM-RoBERTa, the EstBERT model with a shorter sequence length is somewhat better on rubric classification, and the opposite is true for sentiment classification. Overall, the differences between the EstBERT models' performances with both sequence lengths are again relatively small.

5.3 Named Entity Recognition

Table 10 shows the entity-based precision, recall, and F-score of the named entity recognition task. WikiBERT-et model is the best model in this task, obtaining the highest F1-score with both the short and long models and the overall highest F1-score with the short model. XLM-RoBERTa achieves the highest recall in the short model category but remains below the EstBERT in terms of the F1-score. EstBERT, WikiBERT-et and XLM-RoBERTa all benefit from using the smaller sequence length rather than longer, while mBERT shows the opposite behavior.

Table 11 shows the fine-grained scores of each entity type for both the EstBERT, XLM-RoBERTa and the WikiBERT-et shorter model. In alignment with the previous results in Estonian NER (Tkachenko et al., 2013), the prediction of the person entities is the most accurate while the organisation names are the most difficult to predict. The WikiBERT-et is the best on the two most difficult entities ORG and LOC, while the EstBERT model is better than XLM-RoBERTa on these two entities. The WikiBERT-et is notably the best on the most challenging organisation entity, improving the precision over the EstBERT model for almost 4% and over the XLM-RoBERTa for almost 7%, with a considerably smaller loss in recall. One reason for the superiority of

the WikiBERT-et model might lie in the fact that the Wikipedia dataset used to train the WikiBERT-et model probably contains a much higher proportion of organisation names. Although the datasets used to train the other two models also contain the Estonian Wikipedia dataset, it has been diluted in other languages (in case of XLM-RoBERTa) or genres (in case of EstBERT). However, this is just a hypothesis at the moment that has to be studied more in further work.

5.4 Dependency Parsing

Additionally, we also evaluated both the EstBERT, WikiBERT-et and the XLM-RoBERTa models on the Estonian dependency parsing task. The data used in these experiments is the Estonian UDv2.5 described in section 3.1. We trained the parser available in the spaCy Nightly version[9] that also supports transformers. The models were trained with a batch size of 32 and for a maximum of 20K steps, stopping early when the development set performance did not improve for 1600 steps. The parser was trained jointly with a tagger component that predicted the concatenation of POS tags and morphological features. During training, the model was supplied with the gold sentence and token segmentations. During evaluation, the sentence segmentation and tokenisation was done with the out-of-the-box spaCy tokeniser.

The dependency parsing results are in Table 12. In addition to the transformer-based EstBERT and XLM-RoBERTa models, the right-most section also displays the Stanza parser (Qi et al., 2020), trained on the same Estonian UDv2.5 corpus, obtained from the Stanza web page.[10] We add a non-transformer based baseline for this task because dependency parsing was not evaluated

---

Model DepRel	Support	EstBERT Prec	Rec	F1	XLM-RoBERTa Prec	Rec	F1	WikiBERT-et Prec	Rec	F1	Stanza Prec	Rec	F1
UAS		86.07	87.34	86.70	**88.02**	**89.32**	**88.66**	85.97	87.24	86.60	86.69	86.68	86.69
LAS		83.32	84.56	83.94	**85.60**	**86.87**	**86.23**	83.06	84.29	83.67	83.63	83.63	83.63
nmod	4328	81.40	85.65	83.47	**83.73**	**88.49**	**86.05**	80.86	85.42	83.08	82.53	84.27	83.39
obl	4198	80.99	79.78	80.38	**83.65**	**82.66**	**83.15**	80.81	79.47	80.13	79.61	77.78	78.68
advmod	3938	78.01	79.02	78.52	**80.43**	**80.45**	**80.44**	78.08	77.96	78.02	78.93	78.11	78.52
root	3214	90.82	89.61	90.21	**91.88**	**91.91**	**91.90**	90.32	89.73	90.03	90.18	87.46	88.80
nsubj	2682	92.05	93.25	92.65	**93.54**	**94.52**	**94.03**	90.40	92.69	91.53	90.67	89.90	90.28
conj	2476	76.90	78.51	77.70	**81.72**	**83.44**	**82.57**	78.28	78.31	78.30	76.41	78.76	77.57
obj	2437	86.91	88.80	87.84	**88.62**	**90.73**	**89.66**	86.36	87.57	86.96	83.51	84.78	84.14
amod	2411	80.02	84.20	82.05	82.90	86.64	84.73	80.12	83.91	81.97	**91.93**	**89.26**	**90.57**
cc	2029	91.26	90.09	90.67	**92.57**	**91.47**	**92.02**	90.75	89.50	90.12	89.97	88.42	89.19
aux	1372	95.36	95.85	95.60	**95.43**	**95.99**	**95.71**	94.79	95.48	95.13	89.93	95.04	92.42
mark	1277	90.14	90.92	90.53	**92.75**	**93.19**	**92.97**	89.25	89.74	89.50	88.35	89.12	88.73
cop	1202	84.75	87.35	86.03	**85.48**	**87.69**	**86.57**	84.29	87.02	85.63	81.43	86.11	83.70
acl	1063	84.98	85.14	85.06	**86.88**	**87.86**	**87.37**	86.67	85.04	85.85	86.36	80.43	83.29
nsubj:cop	1054	79.98	82.64	81.29	**81.16**	**85.01**	**83.04**	79.34	82.73	81.00	77.78	79.70	78.73
case	908	92.42	92.62	92.52	**93.52**	**93.72**	**93.62**	91.32	92.73	92.02	89.13	91.19	90.15
advcl	857	67.18	65.93	66.55	**73.46**	**71.06**	**72.24**	66.55	66.39	66.47	67.12	63.36	65.19
det	808	83.88	85.02	84.45	**87.89**	**87.13**	**87.51**	82.95	84.28	83.61	80.80	82.80	81.78
parataxis	725	52.96	49.38	51.11	57.50	50.76	53.92	55.59	48.69	51.91	**65.45**	**59.31**	**62.23**
xcomp	641	85.21	87.21	86.20	**88.06**	**88.61**	**88.34**	84.11	86.74	85.41	83.78	83.00	83.39
flat	633	81.44	85.94	83.63	86.64	91.15	88.84	80.09	86.41	83.13	**88.60**	**92.10**	**90.32**
nummod	555	62.88	77.84	69.57	62.00	80.54	70.06	63.12	78.02	69.78	**85.53**	**85.23**	**85.38**
compound:prt	481	86.10	92.72	89.29	**88.20**	**94.80**	**91.38**	85.99	93.14	89.42	85.52	89.60	87.51
appos	376	69.07	71.28	70.16	**74.45**	**80.59**	**77.39**	64.55	73.14	68.58	69.47	72.61	71.00
ccomp	344	82.56	82.56	82.56	**87.03**	**87.79**	**87.41**	80.44	84.88	82.60	81.87	78.78	80.30
acl:relcl	315	80.67	83.49	**82.06**	79.00	80.00	79.50	79.30	79.05	79.17	61.32	82.54	70.37
csubj:cop	121	80.47	85.12	82.73	75.74	85.12	80.16	79.23	85.12	82.07	72.79	**88.43**	79.85
csubj	108	81.51	**89.81**	85.46	80.83	**89.81**	85.09	84.26	84.26	84.26	84.91	83.33	84.11
discourse	47	37.14	55.32	44.44	36.92	51.06	42.86	34.33	48.94	40.35	**81.25**	**55.32**	**65.82**
orphan	44	20.83	11.36	14.71	37.93	**25.00**	**30.14**	20.00	18.18	19.05	**45.00**	20.45	28.12
compound	43	88.10	86.05	87.06	83.33	81.40	82.35	**92.11**	81.40	86.42	88.64	**90.70**	**89.66**
cc:preconj	39	66.67	**71.79**	**69.14**	67.57	64.10	65.79	62.79	69.23	65.85	70.27	66.67	68.42
flat:foreign	37	76.19	43.24	55.17	**87.50**	**56.76**	**68.85**	43.75	18.92	26.42	65.38	45.95	53.97
fixed	31	64.71	70.97	67.69	64.86	**77.42**	70.59	57.50	74.19	64.79	**75.86**	70.97	**73.33**
vocative	9	22.22	22.22	22.22	28.57	22.22	25.00	5.56	11.11	7.41	**30.77**	**44.44**	**36.36**
goeswith	8	**100.00**	12.50	22.22	25.00	12.50	16.67	33.33	**25.00**	**28.57**	0.00	0.00	0.00
dep	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
list	1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 12: Dependency parsing results. The best scores over all models are in **bold**. The best scores comparing the EstBERT, WikiBERT-et and the Stanza models are underlined.

by Kittask et al. (2020). Overall, the XLM-RoBERTa model performs the best, both in terms of the UAS and LAS metrics and the individual dependency relations. This is especially true for dependency relations with larger support in the test set. Although in terms of the UAS and LAS, the EstBERT, WikiBERT-et and Stanza models seem to perform similarly, a closer look into the scores of the individual dependency relations reveals that in most cases, especially with relations of larger support, the EstBERT model performs the best. There are few dependency relations where the Stanza system's predictions are considerably more accurate than the BERT-based models, the most notable of them being the adjectival modifier (amod) and the numerical modifier (nummod). Further analyses are needed to gain more insight into these results.

6 Discussion

This objective of this paper was to describe the process of pretraining the language-specific BERT model for Estonian and to compare its performance with the multilingual BERT models as well as with the smaller Estonian WikiBERT model on several NLP tasks. Overall, the pretrained EstBERT was better than the best multilingual XLM-RoBERTa model on five tasks out of seven: UPOS, XPOS, and morphological tagging, rubric classification, and NER. Only in the sentiment classification and dependency parsing tasks, the XLM-RoBERTa was better. Compared to WikiBERT-et, the EstBERT model was better on six tasks out of seven— the WikiBERT-et model was superior only in the NER task, predicting ORG entities considerably better than any other model. We did not observe any consistent difference between the models of different sequence lengths, although the model with the sequence length

512 was trained longer. It is possible that the shorter model was already trained long enough, and the subsequent training of the longer model did not add any effect in that respect, aside from the fact that it can accept longer input sequences.

One crucial aspect of this work was obtaining a large-enough corpus for pretraining the model. We used the Estonian National Corpus 2017, which was the largest corpus available at the time. A newer and larger version of this corpus—the Estonian National Corpus 2019 (Kallas and Koppel, 2019)—has become available meanwhile. There are also few other resources, such as the Estonian part of the CoNLL 2017 raw data (Ginter et al., 2017) and the Oscar Crawl, which probably partially overlap with each other and with the Estonian National Corpus. Still, these corpora would potentially provide additional data that was currently not used.

Another challenge was related to finding annotated datasets for downstream tasks. While the Estonian UD dataset provides annotations to the common dependency parsing pipeline tasks, datasets for other, especially semantic NLP tasks, are scarce. We adopted the Estonian Valence corpus for two-way text classification. However, the labels of this corpus are semi-automatically derived from user ratings, and thus the quality of these annotations cannot be guaranteed. An Estonian coreference dataset with some simple baseline results in nominal coreference resolution has recently become available (Barbu et al., 2020), which gives further opportunities to test out the EstBERT model in future work.

When preprocessing the data and pretraining the model, we mostly followed the process of training the FinBERT model for Finnish (Virtanen et al., 2019). We also decided to truecase our corpus to reduce the number of capitalised words in the vocabulary. The EstBERT model itself was also pretrained on the truecased corpus. However, when training the task-based models for evaluation, the EstBERT was finetuned on the cased datasets. Thus, truecasing the datasets before finetuning might have a positive effect on the results. In order to verify this, the EstBERT-based task-specific models should be finetuned using the truecased annotated datasets as input, and compared with the results reported in this paper.

Although we did see some improvements with EstBERT compared to XLM-RoBERTa on the smaller model for the NER task, the differences in scores were generally relatively small. However, we have observed that the annotations of this NER dataset are occasionally erroneous, containing, for instance, label sequences (I-PER, I-PER) instead of (B-PER, I-PER). We have also observed unlabelled entities in the text. Thus, the small variations in the systems' results might not be informative about the systems themselves but can instead stem from the noise in the data. Although these annotation errors have been noticeable enough,

the magnitude of these errors has not been quantified.

The differences between the EstBERT and the XLM-RoBERTa model were, in most cases, relatively small. In previous experiments with several multilingual BERT models on the same Estonian tasks (Kittask et al., 2020), the XLM-RoBERTa proved to be the best multilingual model. This suggests that one option to obtain an even better Estonian language-specific model would be to train an Estonian-specific RoBERTa by initializing the model with the parameters of the XLM-RoBERTa. Considering that the multilingual RoBERTa already performs very well on Estonian tasks, finetuning it with more Estonian data would hopefully bias it even more to the Estonian language while also maintaining the gains obtained from multilingualism.

7 Conclusion

We presented EstBERT, the largest BERT model pretrained specifically on the Estonian language. While several existing multilingual BERT models include Estonian, the only language-specific Estonian BERT model available until now has been trained on the relatively small Wikipedia data. In order to pretrain the EstBERT model, we used the largest Estonian text corpus available at the time. The EstBERT model was put to the test by finetuning it for several tasks, including POS and morphological annotations, dependency parsing, named entity recognition, and text classification. On five tasks out of seven, the classifiers based on EstBERT achieved better performance than the models based on multilingual BERT models, although in several cases, the gap with the best-performing multilingual XLM-RoBERTa was relatively small. These results suggest that training a RoBERTa model for Estonian, initialised with the multilingual model's parameters, might be beneficial. On six tasks out of seven, the models based on EstBERT achieved better results than the Estonian BERT model trained on Wikipedia, suggesting that using more textual data for pretraining leads to a more generalisable model.

References

Eduard Barbu, Kadri Muischnek, and Linda Freienthal. 2020. A Study in Estonian Pronominal Coreference Resolution. In *Volume 328: Human Language Technologies – The Baltic Perspective*, Frontiers in Artificial Intelligence and Applications, pages 3–10.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, and Jorge Pérez. 2020. Spanish pre-trained BERT model and evaluation data. *PML4DC at ICLR*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-Lingual Representation Learning at Scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-Lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

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*.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. *arXiv preprint arXiv:2001.06286*.

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.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 Shared Task - Automatically Annotated Raw Texts and Word Embeddings. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

J. Kallas and K. Koppel. 2018. Estonian National Corpus 2017. Center of Estonian Language Resources.

J. Kallas and K. Koppel. 2019. Estional National Corpus 2019. Center of Estonian Language Resources.

Claudia Kittask, Kirill Milintsevich, and Kairit Sirts. 2020. Evaluating multilingual BERT for Estonian. In *Volume 328: Human Language Technologies – The Baltic Perspective*, Frontiers in Artificial Intelligence and Applications, pages 19–26.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. *Computational Linguistics and Intellectual Technologies*, (18):333–339.

Sven Laur, Siim Orasmaa, Dage Särg, and Paul Tammo. 2020. EstNLTK 1.6: Remastered Estonian NLP Pipeline. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7154–7162.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoit Crabbé, Laurent Besacier, and Didier Schwab. 2020. FlauBERT: Unsupervised Language Model Pre-training for French. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2020. What the [MASK]? Making Sense of Language-Specific BERT Models. *arXiv preprint arXiv:2003.02912*.

Hille Pajupuu, Rene Altrov, and Jaan Pajupuu. 2016. Identifying Polarity in Different Text Types. *Folklore*, 64.

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.

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. 2020. Wikibert models: deep transfer learning for many languages. *arXiv preprint arXiv:2006.01538*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Alexander Tkachenko, Timo Petmanson, and Sven Laur. 2013. Named Entity Recognition in Estonian. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 78–83.

University of Tartu. 2018. UT Rocket. share.neic.no.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for Finnish. *arXiv preprint arXiv:1912.07076*.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv preprint arXiv:1912.09582*.