BERT-like Models for Slavic Morpheme Segmentation

Dmitry Morozov[1,2] **Lizaveta Astapenka**[3] **Anna Glazkova**[4,2]
Timur Garipov[1,2] **Olga Lyashevskaya**[5,6]

[1]The Artificial Intelligence Research Center of Novosibirsk State University
[2]Russian National Corpus [3]St. Petersburg State University [4]University of Tyumen
[5]HSE University [6]Vinogradov Russian Language Institute RAS

Abstract

Automatic morpheme segmentation algorithms are applicable in various tasks, such as building tokenizers and language education. For Slavic languages, the development of such algorithms is complicated by the rich derivational capabilities of these languages. Previous research has shown that, on average, these algorithms have already reached expert-level quality. However, a key unresolved issue is the significant decline in performance when segmenting words containing roots not present in the training data. This problem can be partially addressed by using pre-trained language models to better account for word semantics. In this work, we explored the possibility of fine-tuning BERT-like models for morpheme segmentation using data from Belarusian, Czech, and Russian. We found that for Czech and Russian, our models outperform all previously proposed approaches, achieving word-level accuracy of 92.5-95.1%. For Belarusian, this task was addressed for the first time. The best-performing approach for Belarusian was an ensemble of convolutional neural networks with word-level accuracy of 90.45%.

1 Introduction

Morpheme segmentation is the process of dividing a word into substrings — morphemes — which are the smallest indivisible meaningful elements of a language: roots, prefixes, suffixes, and others. This process can be essential for language learning, particularly for languages with rich derivational capabilities. For example, many orthographic rules taught in Russian language school curricula rely on the ability to identify and analyze the internal structure of a word, such as the spelling of voiceless and voiced consonants at the end of prefixes and the verification of unstressed vowels in roots (Volskaya et al., 2018).

Another potential use case for morpheme segmentation is its application as a subword tokenizer for language models. Using a morpheme tokenizer as an alternative to the widely adopted Byte-Pair Encoding (BPE) (Gage, 1994) has been shown by several researchers (Matthews et al., 2018; Nzeyimana and Niyongabo Rubungo, 2022) to improve the quality of trained models. Finally, morpheme annotation is used in large text corpora (Savchuk et al., 2024), which are employed for linguistic research, to enhance user search capabilities.

For all three scenarios mentioned above, the existence of an algorithm for constructing morpheme segmentation — that is, mapping a word form or lemma to its morphemes — is necessary. In some cases, manually compiled and verified dictionaries of morpheme annotations are used for this purpose. This approach is often applied in school education. For example, for the Belarusian language, the School morpheme dictionary of the Belarusian language (Mormysh et al., 2005) is used, while for Russian, the Word Formation Dictionary of the Russian language (Tikhonov, 1990) is employed.

A significant drawback of this approach is the need for constant adaptation to the emergence of new words in the language. For Slavic languages, with their extensive derivational capabilities, expanding and maintaining such dictionaries requires regular, long-term work by expert linguists. Moreover, the lack of a unified interpretation of the term "morpheme segmentation" and clear criteria for identifying morphemes (Iomdin, 2019) makes it impossible to develop an analytical solution.

At the same time, machine learning-based morpheme segmentation algorithms have demonstrated high quality, including for Slavic languages such as Czech (Peters and Martins, 2022) and Russian (Sorokin and Kravtsova, 2018; Peters and Martins, 2022; Morozov et al., 2024). In particular, Morozov et al. (2024) found that for Russian, the quality of automatic annotation on a random sample, in terms of the number of fully correct segmentations, is on par with expert annotation. In

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6795–6815
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

this case, the work of linguists in expanding morpheme dictionaries could be significantly accelerated by creating automatic draft annotations and subsequently correcting and validating them by the experts.

A significant obstacle to this approach, however, is the sharp decline in annotation quality when models encounter words containing roots not present in the training data (out-of-vocabulary roots, OOV roots) (Morozov et al., 2024). A potential solution to this problem could be the use of pre-trained language models. For example, Pranjić et al. (2024) proposed a binary classifier for detecting morpheme boundaries in words based on fine-tuning the Glot500 model (Imani et al., 2023). This approach, however, is likely unsuitable for annotating large dictionaries due to its computational complexity. Meanwhile, Sorokin (2022) used BERT-like models to enrich the feature representation of words, which improved segmentation quality.

Unlike previous work where BERT was used only to obtain word embeddings, in our work we test whether BERT-like models can be fine-tuned for morpheme segmentation, outperform a high-level baseline on Slavic language data, and address the issue of OOV roots. We utilized a CNN ensemble (Sorokin and Kravtsova, 2018), which outperforms other algorithms on Russian language (Morozov et al., 2024) as the baseline. We used four datasets with morpheme annotations: two for Russian and one each for Czech and Belarusian. Russian and Czech were selected as languages with sufficient representation and existing large BERT-like models, while Belarusian was included as a low-resource language. Unlike the problem statement at the SIGMORPHON competition (Batsuren et al., 2022), we considered the problem of surface segmentation, i.e. dividing a word into morphemes without restoring the original form of the morpheme. In addition, based on the potential application of the algorithm in school education we included in the problem the determination of the type of each morpheme.

Our main contributions are as follows:

1. For Czech and Russian, our fine-tuned BERT-like models outperformed the CNN ensemble. We managed to achieve a share of completely correct annotations of 92-95% in the case of a random test sample, and 72-77% in the case of testing on words with roots that were not found in the training sample. The proportion of erroneous segmentations on a random sample decreased by 30-45%, and on words with OOV roots by 9-15%. The results obtained exceed all previously presented results for these languages.

2. For Belarusian, the CNN ensemble showed better performance in both types of testing, achieving word accuracy of 90.5% in the case of random split and 74.8% in the split-by-roots case. To our knowledge, similar experiments have not been conducted for this language before, which allows us to consider the presented result as a new state-of-the-art baseline. We also publish the first publicly available Belarusian morpheme dataset with morpheme type annotation.

3. We found that when testing on words with OOV roots, almost all roots can be divided into two groups: "recognizable" and "completely unknown". The first group includes roots for which words are annotated completely correctly in 100% of cases, while the second group consists of roots for which words are never annotated completely correctly by the model.

2 Related Work

2.1 Morpheme Segmentation

Research on automatic morpheme segmentation varies significantly in terms of problem formulation. This is because morpheme segmentation algorithms are typically developed for specific applications within larger tasks. One criterion that differentiates approaches is the type of segmentation: surface or canonical (Cotterell et al., 2016). Surface segmentation involves segmenting the original string, while canonical segmentation additionally restores the original form of the morpheme. This distinction becomes evident in cases where language rules cause changes at morpheme boundaries. For example, the Belarusian word *абаненцкі* 'subscription (adj.)' is formed by adding the suffix -*ск*- and the ending -*і* to *абанент* 'subscriber', with the resulting combination -*тс*- at the morpheme boundary transforming into -*ц*-. In this case, the surface segmentation of this word might be either *абан-енц-к-і* or *абан-ен-цк-і*, depending on the adopted paradigm, while the canonical segmentation would be *абан-ент-ск-і*.

Another difference lies in the definition of morpheme types. For many tasks, such as building morpheme tokenizers, defining morpheme types is not essential, whereas for others, such as language education, specifying morpheme types is mandatory.

Finally, a third important distinction is the composition of the dataset used. Typically, two types of datasets are considered: lemma datasets and word form datasets. Morpheme dictionaries compiled by linguists usually work with lemmata, while tokenization obviously requires algorithms that also handle word forms.

Most early, relatively effective morpheme segmentation algorithms belong to the Morfessor family (Creutz and Lagus, 2002). This family includes unsupervised and semi-supervised algorithms, which have been tested on a variety of languages, including English, Finnish, German, Turkish, and others. In the SIGMORPHON 2022 competition (Batsuren et al., 2022), which addressed the task of canonical segmentation of word forms without specifying morpheme types, the Morfessor2 algorithm (Smit et al., 2014) was used as one of the baselines and outperformed the other two baselines — ULM (Kudo, 2018) and WordPiece (Schuster and Nakajima, 2012) — for 8 out of 9 languages.

However, the quality of algorithms in this family remains relatively low. For instance, according to the SIGMORPHON 2022 results, Morfessor2 achieved an F-score for correctly predicted morphemes ranging from 9% to 41%, while the best solutions from competition participants exceeded 90% for each language. Among the solutions presented at the competition, the DeepSPIN team (Peters and Martins, 2022) achieved the best results for all 9 languages. Their models rely on LSTM networks with a specific loss function (DeepSPIN-1 and DeepSPIN-2) and the Transformer architecture (DeepSPIN-3). Among other approaches, the solution by the CLUZH team (Wehrli et al., 2022), an ensemble of neural character-level transducers, deserves mention, as it trailed the leader by only a small margin.

As in other areas of natural language processing, there is significant interest in exploring the potential of large language models for solving morpheme segmentation tasks. For example, Pranjić et al. (2024) presented a fine-tuned model for detecting morpheme boundaries in words, which demonstrated superior results for several low-resource languages. A drawback of this work, however, is the high computational complexity of the algorithm, which sequentially iterates through all possible morpheme boundary positions in a word.

2.2 Slavic Morpheme Segmentation

Among the three languages considered in this work, Russian has been the most extensively studied in terms of morpheme segmentation. Several research teams (Sorokin and Kravtsova, 2018; Bolshakova and Sapin, 2019, 2022; Morozov et al., 2024) have explored segmentation algorithms based on gradient boosting over decision trees, convolutional neural networks, LSTM networks, and Transformers networks. In most cases, the authors addressed the task of surface segmentation with morpheme type identification, using a dataset consisting of lemmata. The best results across various experiments involving different morpheme dictionaries were achieved using a CNN ensemble, with the proportion of fully correct annotations reaching 88-90% (Morozov et al., 2024).

In the case of the Czech language, most of the research is related to the DeriNet database[1]. Macháček et al. (2018) investigate the effectiveness of two linguistically uninformed subword construction methods (Byte Pair Encoding and Subword Text Encoder) in handling morphological variations in Czech. Svoboda and Sevcíková (2022) explore the possibility of automatic construction of word-formation chains.

Finally, we were unable to find any relevant studies for Belarusian.

Among the languages represented at SIGMORPHON, Czech and Russian were also included. For Czech, the best result was achieved using the DeepSPIN-2 model (F-score=93.88), while for Russian, the DeepSPIN-3 model yielded the highest performance (F-score=99.35). However, Morozov et al. (2024) demonstrated that the quality of DeepSPIN-3 drops sharply when transitioning to surface segmentation, particularly when dealing with OOV roots. Additionally, the dataset for Russian in SIGMORPHON 2022 consisted of only 10% lemmata and 90% word forms, and according to official results (Batsuren et al., 2022), the performance of the DeepSPIN-3 approach on Russian lemmata drops significantly below an F-score of 93 (a detailed analysis for Czech was not provided in the paper).

[1] https://ufal.mff.cuni.cz/derinet

Morozov et al. (2024) highlighted that the problem of recognizing OOV roots remains a key unresolved issue. While several algorithms achieve 85-90% fully correct annotations when tested on random words, this figure drops to 67-72% when tested on words containing OOV roots. In the case of DeepSPIN-3, testing without morpheme type identification showed 81% fully correct annotations for a random sample of words, with 12% of segmentations not matching the original word letter by letter. However, when tested on words with OOV roots, the proportion of fully correct annotations dropped to 14.5%, with 74% of words having segmentations that did not match the original word letter by letter.

The problem of "recognizing" OOV roots can be partially addressed by using embeddings from BERT-like models, as demonstrated by Sorokin (2022) and Morozov et al. (2024). However, it remains an open question whether it is possible to rely solely on BERT-like models for segmentation without external models, by fine-tuning a BERT-like model specifically for this task.

3 Data

3.1 Belarusian

Since no machine-readable annotated dataset for the Belarusian language could be found, we prepared such a dataset ourselves. As the source of analyses, we used the School morpheme dictionary of the Belarusian language (Mormysh et al., 2005). Since this dictionary does not include annotations for morpheme types, we conducted this annotation ourselves with the involvement of native Belarusian speakers with linguistic education. For annotation, we used five types of morphemes: root (**ROOT**), prefix (**PREF**), suffix (**SUFF**), ending (**END**), and linking vowel (**LINK**). We decided not to separate suffixes and postfixes, since the original dictionary contains only a few dozen examples of the use of postfixes. Zero endings and zero suffixes were excluded from consideration. The final dataset, **Slounik**, contains annotations for 31,057 words. The dataset is available for downloading under the CC-BY-NC-SA 4.0 license[2].

3.2 Czech

For the Czech language, we utilized the **DeriNet 2.1** database[3]. It contains annotations for 1,248,572 words, with three annotated morpheme types: root, prefix, and suffix. For the experiment, we excluded proper nouns from consideration. The final dataset contains annotations for 820,387 words.

3.3 Russian

For the Russian language, we used two datasets previously utilized by Morozov et al. (2024): **Morphodict-T**, based on the "Word Formation Dictionary of the Russian Language" (Tikhonov, 1990), and **Morphodict-K**, based on the "Dictionary of Morphemes of the Russian Language" (Kuznetsova and Efremova, 1986). These datasets contain annotations for 95,895 and 75,649 words, respectively, differing in both their vocabulary and their approach to morpheme segmentation. Morphodict-K uses an approach that emphasizes strong but not maximal splitting of morphemes and parallels to structurally similar words, while Morphodict-T uses the so-called Vinokur criterion, which requires the existence of a corresponding word-formation chain in modern Russian to isolate a morpheme. Both of these datasets use seven types of morphemes: root, prefix, suffix, ending, postfix (**POST**), linking vowel, and hyphen (**HYPH**).

3.4 Datasets Statistics

A brief numerical description of the datasets used is provided in Table 1.

4 Models and Experimental Setup

Our approach relies on character-level annotation. For this purpose, each letter in a word is assigned a label consisting of two elements: the position of the letter within the morpheme (B for beginning, M for middle, E for end, S for single-letter morpheme) and the type of the morpheme itself. Thus, a total of 22 different labels are possible during conversion (LINK and HYPH can only be single-letter). As a result, the Czech word *šimlatost* 'shyness' with the root *šimlat-* and the suffix *-ost* is mapped to the sequence of labels ['B-ROOT', 'M-ROOT', 'M-ROOT', 'M-ROOT', 'M-ROOT', 'E-ROOT', 'B-SUFF', 'M-SUFF', 'E-SUFF']. We considered two approaches. In the first case, the model was fed a sequence of letters, and a sequence of labels was expected as output. In the second case, we

[2] https://huggingface.co/datasets/ruscorpora/morphodict-bel

[3] https://ufal.mff.cuni.cz/derinet

Dataset	**Slounik**	**DeriNet**	**Morphodict-T**	**Morphodict-K**
Language	Belarusian	Czech	Russian	Russian
Unique words	31,057	820,387	95,895	75,649
Unique morphemes	6,350	109,208	15,899	8,079
Unique roots	6,095	102,889	15,253	7,148
Avg morphemes per word	3.74	4.64	3.86	4.12
Avg morpheme occurrence	18.31	34.89	23.29	38.56
Avg root occurrence	5.20	7.97	7.54	12.24
Avg characters in root	4.18	5.30	5.52	4.62
Morpheme types	PREF, ROOT, SUFF, END, LINK	PREF, ROOT, SUFF	PREF, ROOT, SUFF, END, LINK, POST, HYPH	PREF, ROOT, SUFF, END, LINK, POST, HYPH

Table 1: Brief characteristics of the datasets

prefixed the sequence of letters with the word itself, assigning it a special label ‘0’, that is, for the word *šimlatost* the input sequence was equal to the following: [‘šimlatost’, ‘š’, ‘i’, ‘m’, ‘l’, ‘a’, ‘t’, ‘o’, ‘s’, ‘t’], and the output one: [‘0’, ‘B-ROOT’, ‘M-ROOT’, ‘M-ROOT’, ‘M-ROOT’, ‘M-ROOT’, ‘E-ROOT’, ‘B-SUFF’, ‘M-SUFF’, ‘E-SUFF’].

We conducted two series of experiments: one with random dataset train-test splitting and another with splitting by roots. In both cases, we used 5-fold cross-validation. In the first case, each dataset was randomly divided into 5 nearly equal folds. In the second case, we divided all roots mentioned in the dataset into 5 nearly equal groups and then included all words with roots from group k in fold k. In this scenario, words containing two or more roots were excluded from consideration. The sizes of the folds in the second case were quite close but not equal (see Table 2).

Dataset	**Min fold size**	**Max fold size**
Slounik	5,629	6,712
DeriNet	157,380	170,613
Morphodict-T	15,253	16,013
Morphodict-K	12,401	13,667

Table 2: Size of the folds in the split-by-roots experiment (in words)

To evaluate the quality of annotation, we used metrics previously employed in (Sorokin and Kravtsova, 2018; Morozov et al., 2024): F-score, Precision, and Recall for morpheme boundaries; F-score, Precision, and Recall for root boundaries; character-level Accuracy; and the proportion of completely correct annotations (WordAccuracy). In the case where the word itself was prefixed to the sequence of letters, the label assigned to it was not taken into account when calculating the metrics.

In experiments with a CNN ensemble, we used the implementation by Sorokin and Kravtsova (2018)[4]. We used 3 models in the ensemble, with the window size set to 5. Each model was trained for a maximum of 25 epochs, incorporating early stopping with a patience of 10 epochs to prevent overfitting. Model training was performed on an AMD Ryzen 5 5600X CPU without the use of a GPU. Each training epoch took up to 90 seconds (in the case of the DeriNet dataset).

For implementation BERT-like models, we used the `simpletransformers`[5] framework. For Russian and Czech, we selected models that have demonstrated high quality in other tasks: `ruRoberta-large`[6] (355M parameters) (Zmitrovich et al., 2024) and `Czert-B-base-cased`[7] (110M parameters) (Sido et al., 2021), respectively. For Belarusian, we were unable to find a model with similar characteristics. Therefore, we decided to conduct experiments with two models: a smaller one specifically tailored for Belarusian, `roberta-small-belarusian`[8] (16M parameters),

[4] https://github.com/AlexeySorokin/NeuralMorphemeSegmentation

[5] https://simpletransformers.ai/

[6] https://huggingface.co/ai-forever/ruRoberta-large

[7] https://huggingface.co/UWB-AIR/Czert-B-base-cased

[8] https://huggingface.co/KoichiYasuoka/roberta-small-belarusian

and SlavicBERT[9] (180M parameters) (Arkhipov et al., 2019), which was trained on four other Slavic languages: Bulgarian, Czech, Polish, and Russian.

The batch size during training was set to 16, and the learning rate was set to 4e-6. The values of the remaining parameters were set to default. We fine-tuned the BERT-like models for 30 epochs. The models were trained on an Nvidia RTX 4090 GPU. Each training epoch took up to 18 minutes (in the case of the DeriNet dataset). The fine-tuned models are available on request from the corresponding author.

5 Results

The averaged cross-validation results are presented in Tables 3, 4. Full experimental results for each dateset, model, and fold can be found in Appendix E. The best results for each dataset are underlined. F stands for F-score, P stands for Precision, R stands for Recall, Acc stands for character-level accuracy, and WA stands for word-level accuracy. The "+lex" models correspond to models trained with the lemma added to the input sequence.

For three out of the four datasets, BERT-like models lead across all metrics. In the case of random split, the best results were achieved for Czech: 95.4% fully correct annotations and 98.8% character-level accuracy. For all datasets, a WordAccuracy greater than 90% and a character-level accuracy greater than 95% were achieved. When transitioning to testing on OOV roots, the annotation quality, as expected, decreases. However, similar to the random split scenario, BERT-like models, particularly those trained with the addition of the word lemma in the input sequence, outperformed others for Czech and Russian datasets. The best result was achieved on the Morphodict-T dataset: 77.2% fully correct segmetations and 92.82% character-level accuracy. The superiority over the CNN ensemble for Czech and Russian ranged from 2.2 to 4.5 percentage points in terms of WordAccuracy.

For Belarusian, the best result after the CNN ensemble in the random split scenario was achieved by the roberta-small-belarusian model, trained without lemmata in the input sequence. This demonstrates that a model 10 times smaller, trained on a low-resource language, can be more effective for the given task than a larger model trained on related languages. In the split-by-roots case, the SlavicBert model, similarly trained without lemmata, proved competitive: it outperformed CNN for 4 out of 8 metrics and trailed only slightly for the remaining four.

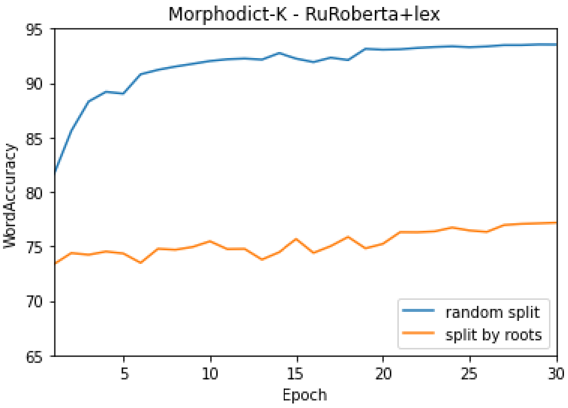

Figure 1: WordAccuracy values during model training

Four out of the five tested models showed improved quality in the split-by-roots scenario when lemmata were added to the training data. The exception was the SlavicBert model, which notably lacked the target language in its training data. Thus, it can be concluded that for models specifically trained on the target language, adding lemmata to the input sequence enhances the model's ability to annotate OOV roots. However, for two of these four models, adding lemmata reduced recognition quality in the random split scenario.

Comparing Precision and Recall for all morphemes and for roots only, it can be observed that Recall for all morphemes is higher than Precision across all datasets and models. For root morphemes, a similar pattern is observed for the CNN ensemble, whereas for BERT-like models with added lemmata, the situation is reversed.

Finally, the analysis of WordAccuracy dynamics during training showed that for random split the value steadily increases, especially in the first 10 epochs. At the same time, for split-by-roots the changes are much less pronounced. An example of WordAccuracy dynamics for the Morphodict-K dataset and the RuRoberta+lex model is shown in Figure 1.

6 Error Analysis

A selective analysis of errors made by the models revealed that two types of deviations from the target annotation occur most frequently: differences in suffix segmentation and differences in extract-

[9] https://huggingface.co/DeepPavlov/bert-base-bg-cs-pl-ru-cased

Model	All morphemes boundaries			Only roots boundaries			Acc	WA
	F	P	R	F	P	R		
Slounik								
CNN	98.41	98.10	98.73	95.26	95.30	95.21	96.95	90.45
Roberta-bel	98.40	98.25	98.55	94.92	95.08	95.08	96.82	90.32
Roberta-bel+lex	98.09	97.88	98.30	94.01	94.13	93.88	96.21	88.86
SlavicBert	98.00	97.72	98.28	93.56	93.72	93.41	95.99	87.73
SlavicBert+lex	97.96	97.68	98.24	93.33	93.58	93.08	95.89	87.58
DeriNet								
CNN	98.91	98.72	99.11	94.09	94.04	94.14	97.45	91.09
Czert	99.40	99.27	99.53	96.63	96.64	96.63	98.69	95.12
Czert+lex	99.44	99.33	99.55	96.78	96.79	96.78	98.76	95.39
Morphodict-T								
CNN	98.09	97.79	98.38	94.08	94.19	93.99	96.61	88.49
ruRoberta	98.56	98.57	98.56	95.31	95.45	95.18	97.39	91.09
ruRoberta+lex	98.76	98.69	98.84	96.07	96.13	96.00	97.78	92.47
Morphodict-K								
CNN	98.66	98.58	98.74	96.24	96.26	96.22	97.40	90.82
ruRoberta	99.09	99.05	99.14	97.43	97.44	97.43	98.19	93.61
ruRoberta+lex	99.10	99.04	99.17	97.36	97.37	97.35	98.19	93.54

Table 3: Average metric value during cross-validation, random split

Model	All morphemes boundaries			Only roots boundaries			Acc	WA
	F	P	R	F	P	R		
Slounik								
CNN	95.44	94.88	96.00	83.98	83.97	83.98	91.24	74.78
Roberta-bel	95.36	95.51	95.21	83.60	83.67	83.53	90.80	73.55
Roberta-bel+lex	95.48	95.48	95.49	83.71	83.79	83.64	91.12	74.59
Slavic	95.59	95.29	95.89	84.28	84.49	84.06	91.17	74.74
Slavic+lex	95.49	95.49	95.73	83.70	83.84	83.56	91.03	74.33
DeriNet								
CNN	96.30	96.42	96.18	80.07	79.98	80.16	91.38	70.35
Czert	96.32	96.91	95.74	79.26	79.31	79.22	91.42	70.39
Czert+lex	96.62	97.06	96.18	80.70	80.74	80.66	92.08	72.63
Morphodict-T								
CNN	94.71	94.46	94.46	81.97	81.96	81.98	90.16	70.53
ruRoberta	94.98	94.67	95.31	82.08	81.91	82.26	90.15	71.10
ruRoberta+lex	95.59	96.09	95.08	84.34	84.37	84.30	91.70	74.95
Morphodict-K								
CNN	95.19	95.35	95.04	82.50	82.46	82.54	91.30	72.63
ruRoberta	95.26	95.78	94.74	82.32	82.37	82.28	91.39	73.32
ruRoberta+lex	96.09	96.38	95.81	84.87	84.95	84.78	92.82	77.17

Table 4: Average metric value during cross-validation, split-by-roots

ing suffixes and prefixes from the root, with the latter becoming more frequent when transitioning from random split to split-by-roots. Some errors can be explained by mistakes and inconsistencies in dataset annotation (Table 5). This aligns with previously obtained results for the Russian language (Sorokin and Kravtsova, 2018; Bolshakova and Sapin, 2019; Morozov et al., 2024).

The results of analyzing errors made by the models in the split-by-roots experiment proved to be

Dataset	Lemma	Dataset segmentation	Corrected segmentation
Slounik	усынавіцель	у-сын-ав-i:LINK-цель	у-сын-ав-i:SUFF-цель
DeriNet	pedopsychiatricky	pedop-sychiatrick-y	pedo-psychiatrick-y
Morphodict-T	очевидный	очевидн-ый	очевид-н-ый
Morphodict-K	растение	рас-т-ени-е	раст-ени-е

Table 5: Examples of mis-segmentation in datasets

particularly interesting. For each OOV root, we calculated the *root recognition*: the proportion of completely correctly segmented words among all words containing that root. We found that the recognition rate of the vast majority of roots is either 1 or 0 across all models and datasets (Figure 2). The roots of the first type we termed “recognizable”, while the roots of the second type were labeled “completely unknown”. We found two features that differed between these root groups:

1. **Proximity to roots in the training sample.** For example, for the Morphodict-K dataset, the average minimal Levenshtein distance between recognizable roots and training roots was 1.1, versus 1.5 for completely unknown roots.

2. **Shared derivation patterns.** When masking roots in Morphodict-K (e.g. “игр:ROOT/а:SUFF/ть:END” ‘to play’ → “[MASK]:ROOT/а:SUFF/ть:END”), 90% of words with recognizable roots had a training-set word with an identical masked pattern, compared to only 73% for completely unknown roots. A similar trend holds for Belarusian – e.g., the erroneous segmentation “грукат:ROOT/а:SUFF/ць:SUFF” ‘to rumble’ (vs. the correct “грук:ROOT/ат:SUFF/а:SUFF/ць:SUFF”) likely arises because the pattern “ат:ROOT/а:SUFF/ць:SUFF” appears 46 times in the training set, while “/ат:SUFF/а:SUFF/ць:SUFF” occurs only 7 times.

Another interesting observation is that the segmentation behavior (including root recognizability) is stable across random seeds. For Morphodict-K, three model reruns produced identical segmentations in 90% of cases (76% of segmentations were fully correct for each of the models, 17% were incorrect for each of the models). The overlap of “recognizable root” sets across models reached 95% of each set’s size.

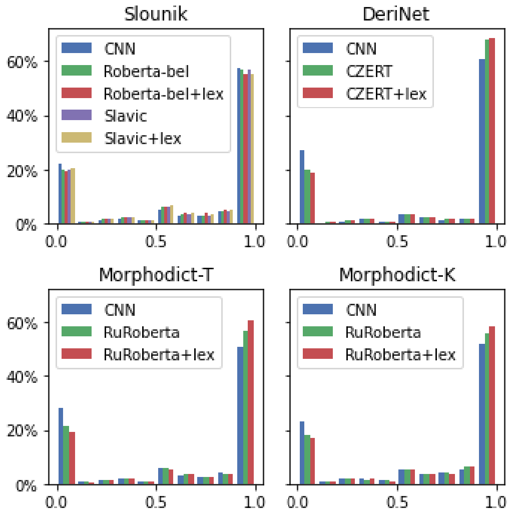

Figure 2: Histogram for the root recognition

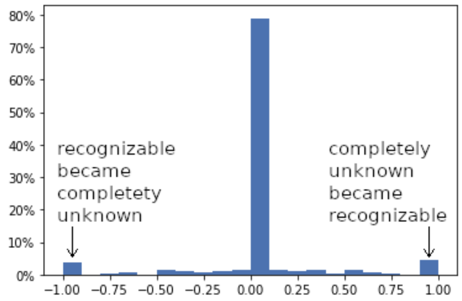

Figure 3: Histogram of the difference in root recognition for the Czert and Czert+lex models

When comparing this statistic for models trained without adding lemmata to the input sequence versus those trained with lemmata, it turns out that roots most often do not transition between categories. However, for each pair of models, there are roots that transition from “completely unknown” to “recognizable” and vice versa. An example of a histogram of such difference for the Czert models before and after adding a lemma to the input sequence is shown in Figure 3. Unfortunately, we were unable to identify clear patterns or dependencies between the roots and this dynamic.

7 Comparison with DeepSPIN-3

The models presented at the SIGMORPHON 2022 cannot be directly compared to our approach due to differences in task formulation: the competition focused on canonical segmentation, whereas our work addresses surface segmentation. Nevertheless, the exceptionally high performance achieved by the DeepSPIN-3 algorithm during the competition prompted our interest in conducting at least an approximate comparison. Beyond a direct performance comparison, this would also help assess the applicability of algorithms designed for canonical segmentation task to the surface one.

Since DeepSPIN-3 does not involve morpheme type labeling, types were removed. The results are presented in Table 6. In addition to the Levenshtein distance used in the competition, we included WordAccuracy — the proportion of fully correct segmentations.

Dataset	Levenshtein	WordAccuracy
	random split	
Slounik	0.80	68.65
DeriNet	0.10	92.28
Morphodict-T	0.41	79.69
Morphodict-K	0.55	77.10
	split by roots	
Slounik	2.22	12.11
DeriNet	0.68	55.16
Morphodict-T	1.58	25.85
Morphodict-K	1.90	13.38

Table 6: Results obtained using the DeepSPIN-3 algorithm

The results demonstrate that in most scenarios DeepSPIN-3 underperforms not only pretrained models but also the CNN ensemble baseline. The sole exception is DeriNet dataset in the random-split scenario, where DeepSPIN-3 achieved 92% word-level accuracy (vs. 91% for CNN ensemble and 95% for pretrained models). For other languages, DeepSPIN-3 is significantly inferior, which we attribute to the size of the training sets.

The performance drops drastically in the split-by-roots scenario — the average Levenshtein distance increases by 0.5-1.5, with word-level accuracy decreasing by more than 30 percentage points. Draft error analysis reveals that the algorithm consistently generates roots similar to training set instances. For example, in Morphodict-K, the word “отплатить” ‘to pay back’ with correct segmentation от:PREF/плат:ROOT/и:SUFF/ть:END receives incorrect segmentation от:PREF/лат:ROOT/и:SUFF/ть:END (missing the root letter -п-), where лат:ROOT appears in training set words like “подлатать” ‘to patch up’. Our analysis shows that generating Levenshtein-close training set roots accounts for two-thirds of all errors. A potential solution might involve restricting model generation to non-root morphemes while treating unmarked segments as roots, though this would require substantial algorithm modifications and warrants separate investigation.

8 Conclusion

In this work, we investigated the applicability of BERT-like models to the task of morpheme segmentation using the material of the Belarusian, Czech, and Russian languages. We used a CNN ensemble as a baseline. For Czech and Russian, our fine-tuned BERT-like models outperformed the baseline. We managed to achieve a share of completely correct annotations of 92-95% in the case of a random test sample, and 72-77% in the case of testing on words with roots that were not found in the training sample. The proportion of incorrect segmentations on a random sample decreased by 30-45%, and on words with OOV roots by 9-15%. The results obtained exceed all previously presented results for these languages.

For Belarusian, the CNN ensemble showed better performance in both types of testing, achieving word accuracy of 90.5% in the case of random split and 74.8% in the split-by-roots case. To our knowledge, similar experiments have not been conducted for this language before, which allows us to consider the presented result as a new state-of-the-art baseline. We also prepared the first publicly available Belarusian morpheme dataset with morpheme type annotation.

We also found that when testing on words with OOV roots, almost all roots can be divided into two groups: “recognizable” and “completely unknown”. The first group includes roots for which words are annotated completely correctly in 100% of cases, while the second group consists of roots for which words are never annotated completely correctly by the model.

9 Limitations

1. **Significant computational power required for training and inference BERT-like models.** Unlike the CNN ensemble, fine-tuning and inference of BERT-like models require the use of sufficiently powerful GPUs (albeit from the consumer segment). On the other hand, preliminary testing indicated that using large language models on this task would require significantly more computational time and resources, so we decided not to conduct full-scale experiments with LLMs. However, automatic expansion of morpheme dictionaries does not require frequent model runs, since new words appear in the language relatively rarely. In this case, the additional costs of BERT-like models can be compensated by higher quality of the labeling.

2. **Separation of roots into "recognizable" and "completely unknown" requires additional linguistic research.** We found that for both CNN and BERT-like approaches, when testing on words with OOV roots, two categories of roots emerge: "recognizable" and "completely unknown". However, a detailed analysis of why certain roots fall into one category or another is planned for future work.

3. **Lack of pre-trained Belarusian models and small size of the morpheme dataset.** Although we were able to draw several conclusions about morpheme segmentation for the Belarusian language, the lack of relatively large pre-trained models for this language does not allow us to consider CNN as the unequivocal leader. On the contrary, the results obtained for Czech and Russian suggest that the absence of models and the small size of the dataset were the reasons for the failure of the BERT-like approach. With the emergence of larger Belarusian models and annotated datasets, our study should be replicated to draw more reliable conclusions.

Acknowledgments

This work was supported by a grant for research centers, provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement with the Novosibirsk State University dated April 17, 2025 No. 139-15-2025-006: IGK 000000C313925P3S0002.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Elena Bolshakova and Alexander Sapin. 2019. Bi-LSTM model for morpheme segmentation of Russian words. In *Artificial Intelligence and Natural Language*, pages 151–160, Cham. Springer International Publishing.

Elena I. Bolshakova and Alexander S. Sapin. 2022. Building a combined morphological model for russian word forms. In *Analysis of Images, Social Networks and Texts*, pages 45–55, Cham. Springer International Publishing.

Francois Chollet et al. 2015. Keras.

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. A joint model of orthography and morphological segmentation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 664–669, San Diego, California. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to

500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

B. L. Iomdin. 2019. How to define words with the same root? *Russian Speech = Russkaya Rech'*, (1):109–115.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

A. I. Kuznetsova and T. F. Efremova. 1986. *Dictionary of Morphemes of the Russian Language*. Russkii yazyk, Moscow.

Dominik Macháček, Jonáš Vidra, and Ondřej Bojar. 2018. Morphological and language-agnostic word segmentation for NMT. In *International Conference on Text, Speech, and Dialogue*, pages 277–284. Springer.

Austin Matthews, Graham Neubig, and Chris Dyer. 2018. Using morphological knowledge in open-vocabulary neural language models. 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 1435–1445, New Orleans, Louisiana. Association for Computational Linguistics.

L. S. Mormysh, A. M. Bordovich, and L. M. Shakun. 2005. *School morpheme dictionary of the Belarusian language [SHkol'ny marfemny slovnik belaruskaj movy]*. Aversev, Minsk.

Dmitry Morozov, Timur Garipov, Olga Lyashevskaya, Svetlana Savchuk, Boris Iomdin, and Anna Glazkova. 2024. Automatic morpheme segmentation for Russian: Can an algorithm replace experts? *Journal of Language and Education*, 10(4):71–84.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. KinyaBERT: a morphology-aware Kinyarwanda language model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363, Dublin, Ireland. Association for Computational Linguistics.

Ben Peters and Andre F. T. Martins. 2022. Beyond characters: Subword-level morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 131–138, Seattle, Washington. Association for Computational Linguistics.

Marko Pranjić, Marko Robnik-Šikonja, and Senja Pollak. 2024. LLMSegm: Surface-level morphological segmentation using large language model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10665–10674, Torino, Italia. ELRA and ICCL.

Svetlana O Savchuk, Timofey Arkhangelskiy, Anastasiya A Bonch-Osmolovskaya, Ol'ga V Donina, Yuliya N Kuznetsova, Ol'ga N Lyashevskaya, Boris V Orekhov, and Mariya V Podryadchikova. 2024. Russian National Corpus 2.0: New opportunities and development prospects. *Voprosy Jazykoznanija*, (2):7–34.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Jakub Sido, Ondřej Pražák, Pavel Přibáň, Jan Pašek, Michal Seják, and Miloslav Konopík. 2021. Czert – Czech BERT-like model for language representation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1326–1338, Held Online. INCOMA Ltd.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Alexey Sorokin. 2022. Improving morpheme segmentation using bert embeddings. In *Analysis of Images, Social Networks and Texts*, pages 148–161, Cham. Springer International Publishing.

Alexey Sorokin and Anastasia Kravtsova. 2018. Deep convolutional networks for supervised morpheme segmentation of Russian language. In *Artificial Intelligence and Natural Language*, pages 3–10, Cham. Springer International Publishing.

Emil Svoboda and Magda Sevcíková. 2022. Word formation analyzer for czech: Automatic parent retrieval and classification of word formation processes. *The Prague Bulletin of Mathematical Linguistics*, 118:55.

A. N. Tikhonov. 1990. *Word Formation Dictionary of the Russian language [Slovoobrazovatel'nyi slovar' russkogo yazyka]*. Russkiy yazyk, Moscow.

Anastasiya S. Volskaya, Tatyana A. Korneyeva, and Tatyana D. Markova. 2018. Word-formation, morphemic, etymological analysis in school. *Amazonia Investiga*, 7(15):190–195.

Silvan Wehrli, Simon Clematide, and Peter Makarov. 2022. CLUZH at SIGMORPHON 2022 shared tasks on morpheme segmentation and inflection generation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 212–219, Seattle, Washington. Association for Computational Linguistics.

Dmitry Zmitrovich, Aleksandr Abramov, Andrey Kalmykov, Vitaly Kadulin, Maria Tikhonova, Ekaterina Taktasheva, Danil Astafurov, Mark Baushenko, Artem Snegirev, Tatiana Shavrina, Sergei S. Markov, Vladislav Mikhailov, and Alena Fenogenova. 2024. A family of pretrained transformer language models for Russian. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 507–524, Torino, Italia. ELRA and ICCL.

A Risks

Our research is primarily foundational. Despite this, we can assume some risks associated with the integration of our algorithm into language learning processes. Although the obtained labeling quality is quite high, the probability of errors is still significant. The segmentations generated by the model must be validated by a human expert before being implemented in educational materials.

In addition, the datasets used may contain a small number of examples of obscene and offensive vocabulary. However, all the data used are based on academic dictionaries and projects, so we believe that such words are an integral part of the language, and morpheme segmentation models should be able to work with such words.

B Scientific Artifacts

Our work uses external datasets, pre-trained models and software libraries. We created one scientific artifact, the Slounik dataset[10], publicly available under the CC-BY-NC-SA 4.0 license.

B.1 Datasets

The datasets we utilized do not contain any personal information. The datasets may contain a small number of examples of obscene and offensive vocabulary. However, all the data used are based on academic dictionaries and projects, so we believe that such words are an integral part of the language, and morpheme segmentation models should be able to work with such words.

1. **Slounik**. As a basis for preparing the Slounik dataset, we used the version of the School morpheme dictionary of the Belarusian language (Mormysh et al., 2005) available for non-commercial use in the repository of the Belarusian State Pedagogical University named after Maxim Tank (BSPU)[11][12][13][14][15][16][17]. According to the description of the repository, it is allowed to copy and quote materials exclusively for non-commercial purposes with the obligatory indication of the author of the work and a hyperlink to the BSPU Repository. We translated the dictionary into a machine-readable form. During the additional labeling, the annotators did not make changes to the morpheme segmentation, but only marked the types of segmented morphemes. The Slounik dataset[18] is publicly available under the CC-BY-NC-SA 4.0 license, including mandatory mention of the authors of the original dictionary and a link to the BSPU repository.
2. **DeriNet**. Adapted from the DeriNet[19] dataset, which is available in the LINDAT/CLARIAH-CZ digital library[20] at the Institute of Farmal and Applied Linguistics, Faculty of Mathematics and Physics, Charles university under the terms of the CC-BY-NC-SA 3.0 license.
3. **Morphodict-T** and **Morphodict-K**. The datasets were provided to us for use exclusively for scientific purposes under a license agreement with the Russian National Corpus[21] (Savchuk et al., 2024). A detailed description of the differences in the markup between the datasets can be found on the Corpus website[22].

B.2 Pre-trained Models

We utilized four pre-trained models from HuggingFace[23]:

1. `roberta-small-belarusian`[24]. This model is monolingual (Belarusian), has 16M parameters, and is available under the CC-BY-SA 4.0 license.

[10] https://huggingface.co/datasets/ruscorpora/morphodict-bel

[11] http://elib.bspu.by/handle/doc/30574

[12] http://elib.bspu.by/handle/doc/30575

[13] http://elib.bspu.by/handle/doc/30576

[14] http://elib.bspu.by/handle/doc/30577

[15] http://elib.bspu.by/handle/doc/30578

[16] http://elib.bspu.by/handle/doc/30579

[17] http://elib.bspu.by/handle/doc/30580

[18] https://huggingface.co/datasets/ruscorpora/morphodict-bel

[19] https://ufal.mff.cuni.cz/derinet

[20] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3765

[21] https://ruscorpora.ru/en

[22] https://ruscorpora.ru/en/page/instruction-derivation

[23] https://huggingface.co/

[24] https://huggingface.co/KoichiYasuoka/roberta-small-belarusian

2. SlavicBERT[25] (Arkhipov et al., 2019). This model is multilingual (Bulgarian, Czech, Polish, and Russian), has 180M parameters, and is available under the Apache 2.0 license.

3. Czert-B-base-cased[26] (Sido et al., 2021). This model is monolingual (Czech), has 110M parameters, and is available under the CC-BY-NC-SA 4.0 license.

4. ruRoberta-large[27] (Zmitrovich et al., 2024). This model is monolingual (Russian), has 355M parameters, and is available under the MIT license.

B.3 Software libraries

1. In experiments with a CNN ensemble, we used the implementation by Sorokin and Kravtsova (2018) from publicly available repository[28]. Unfortunately, there is no indication of licensing terms in the repository. The code in the repository utilizes the keras (Chollet et al., 2015) and tensorflow (Abadi et al., 2016) libraries. We used version 2.12.0 of both libraries to run it. We used the following parameters set by the configuration file: models number — 3, number of convolutional layers — 3, window size — 5, filters number — 192, dense output units — 64, validation split ratio — 0.2, dropout ratio — 0.2.

2. For implementation BERT-like models, we used the simpletransformers[29] framework, which is available under Apache 2.0 license. We used the NERModel class to load and fine-tune the models. All models were loaded via HuggingFace API[30]. The batch size during training was set to 16, and the learning rate was set to 4e-6. The values of the remaining parameters were set to default.

C Computational Experiments

Training all 20 convolutional neural network ensembles together took less than 48 hours on an AMD Ryzen 5 5600X CPU.

BERT-like models fine-tuning was performed on a single Nvidia RTX 4090 GPU. Training time depended on the specific dataset and, to a lesser extent, the base model. Fine-tuning of a single roberta-small-belarusian model on the Slounik dataset took less than one hour, while fine-tuning of the Czert model on the DeriNet dataset required about nine hours. In total, 160 GPU hours were used for fine-tuning the models, including preliminary experiments.

D Human Annotation

We used human annotators only for additional labeling of the Slounik dataset. It should be noted that during this process we did not change the segmentation of morphemes in the source dictionary. The task of type labeling is not complex, since the main challenge in morpheme segmentation of Belarusian (as well as other Slavic languages) is precisely the division of the source string. In the vast majority of cases, the morpheme type is easily determined unambiguously: prefixes, suffixes, endings and connecting vowels can only be from a fixed set of strings, and the intersection between the set of roots and the rest of the morphemes is extremely small (no more than two dozen strings). In order to avoid potential discrepancies in labeling between annotators, we decided not to separate postfixes from suffixes, labeling all such morphemes as suffixes.

Preliminary labeling was carried out by a native Russian speaker (as a morphologically close language) with linguistic background, and validation of the labeling results was carried out by two Russian- and Belarusian-speaking annotators with linguistic education.

The annotators were instructed to assign each morpheme of the word one of five possible types: PREF (prefix), ROOT, SUFF (suffix), END (ending), LINK (linking vowel). The primary annotator received data as a json dictionary. Each lemma in the dictionary corresponded to a list of several morphemes. The concatenation of morphemes coincided with the original lemma. The annotator's task was to add the type to each morpheme, separating it with the ":" character, for example, for the word *абавязаць*, the annotator received the list ["абавяз", "а", "ць"], to which the annotator matched the list ["абавяз:ROOT", "а:SUFF", "ць:SUFF"]. The annotators were informed that the labeled dataset would be used, among other

[25] https://huggingface.co/DeepPavlov/bert-base-bg-cs-pl-ru-cased

[26] https://huggingface.co/UWB-AIR/Czert-B-base-cased

[27] https://huggingface.co/ai-forever/ruRoberta-large

[28] https://github.com/AlexeySorokin/NeuralMorphemeSegmentation

[29] https://simpletransformers.ai/

[30] https://huggingface.co/

things, to train morpheme segmentation models.

E Detailed Experimental Results

The results obtained during the experiments for each of the folds are presented in Tables 7, 11, and 12 for the Slounik dataset, in Tables 8 and 13 for the DeriNet dataset, in Tables 9 and 14 for the Morphodict-T dataset, and in Tables 10 and 15 for the Morphodict-K dataset. Additionally, the average value and standard deviation are presented.

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	StdDev
random split							
$\text{Precision}_{\text{all}}$	97.92	98.20	98.24	98.09	98.04	98.10	0.13
$\text{Recall}_{\text{all}}$	98.85	98.65	98.47	98.89	98.79	98.73	0.17
F1_{all}	98.39	98.43	98.36	98.49	98.41	98.41	0.05
$\text{Precision}_{\text{root}}$	95.03	95.36	95.12	95.60	95.39	95.30	0.23
$\text{Recall}_{\text{root}}$	95.08	95.29	95.03	95.42	95.26	95.21	0.16
F1_{root}	95.05	95.32	95.08	95.51	95.32	95.26	0.19
Accuracy	96.85	96.96	96.93	97.11	96.90	96.95	0.10
WordAccuracy	90.41	90.23	90.18	90.87	90.55	90.45	0.28
split by roots							
$\text{Precision}_{\text{all}}$	94.90	94.62	94.63	94.66	95.59	94.88	0.41
$\text{Recall}_{\text{all}}$	96.01	96.14	96.20	95.73	95.93	96.00	0.18
F1_{all}	95.45	95.37	95.41	95.20	95.76	95.44	0.20
$\text{Precision}_{\text{root}}$	85.09	83.19	84.60	82.52	84.47	83.97	1.07
$\text{Recall}_{\text{root}}$	85.09	83.19	84.60	82.52	84.51	83.98	1.08
F1_{root}	85.09	83.19	84.60	82.52	84.49	83.98	1.08
Accuracy	91.56	91.12	91.00	90.80	91.74	91.24	0.39
WordAccuracy	75.41	73.56	74.65	73.78	76.49	74.78	1.21

Table 7: Segmentation quality of the Slounik dataset using a CNN ensemble

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	StdDev
random split							
$\text{Precision}_{\text{all}}$	98.63	98.80	98.90	98.59	98.70	98.72	0.11
$\text{Recall}_{\text{all}}$	99.20	99.07	98.96	99.17	99.12	99.11	0.09
F1_{all}	98.92	98.94	98.93	98.88	98.91	98.91	0.02
$\text{Precision}_{\text{root}}$	94.08	94.18	94.07	93.80	94.07	94.04	0.13
$\text{Recall}_{\text{root}}$	94.17	94.27	94.19	93.90	94.17	94.14	0.13
F1_{root}	94.12	94.22	94.13	93.85	94.12	94.09	0.13
Accuracy	97.50	97.50	97.46	97.38	97.44	97.45	0.04
WordAccuracy	91.10	91.28	91.15	90.86	91.06	91.09	0.13
split by roots							
$\text{Precision}_{\text{all}}$	96.10	96.49	96.17	96.71	96.63	96.42	0.24
$\text{Recall}_{\text{all}}$	96.68	95.92	96.26	95.85	96.21	96.18	0.29
F1_{all}	96.39	96.20	96.22	96.28	96.42	96.30	0.09
$\text{Precision}_{\text{root}}$	80.07	79.70	80.36	79.59	80.20	79.98	0.30
$\text{Recall}_{\text{root}}$	80.13	79.89	80.57	79.79	80.40	80.16	0.30
F1_{root}	80.10	79.79	80.47	79.69	80.30	80.07	0.30
Accuracy	91.47	91.29	91.36	91.39	91.38	91.38	0.06
WordAccuracy	70.52	69.85	70.16	70.72	70.51	70.35	0.31

Table 8: Segmentation quality of the DeriNet dataset using a CNN ensemble

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	StdDev
random split							
Precision$_{\text{all}}$	97.79	97.76	97.76	97.64	98.00	97.79	0.13
Recall$_{\text{all}}$	98.38	98.33	98.40	98.46	98.32	98.38	0.06
F1$_{\text{all}}$	98.09	98.04	98.09	98.05	98.16	98.09	0.05
Precision$_{\text{root}}$	94.02	94.17	94.07	93.98	94.69	94.19	0.29
Recall$_{\text{root}}$	93.80	93.97	93.88	93.93	94.35	93.99	0.21
F1$_{\text{root}}$	93.91	94.07	93.97	93.95	94.52	94.08	0.25
Accuracy	96.57	96.55	96.63	96.53	96.77	96.61	0.10
WordAccuracy	88.22	88.35	88.56	88.39	88.94	88.49	0.28
split by roots							
Precision$_{\text{all}}$	94.45	95.13	93.92	94.73	94.05	94.46	0.50
Recall$_{\text{all}}$	95.46	95.20	94.78	94.28	95.09	94.96	0.45
F1$_{\text{all}}$	94.95	95.16	94.35	94.51	94.57	94.71	0.34
Precision$_{\text{root}}$	83.03	82.14	81.89	81.74	81.01	81.96	0.73
Recall$_{\text{root}}$	83.04	82.16	81.90	81.74	81.07	81.98	0.72
F1$_{\text{root}}$	83.03	82.15	81.90	81.74	81.04	81.97	0.72
Accuracy	90.53	90.87	89.52	89.95	89.95	90.16	0.53
WordAccuracy	72.04	72.61	68.24	69.67	70.07	70.53	1.79

Table 9: Segmentation quality of the Morphodict-T dataset using a CNN ensemble

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	StdDev
random split							
Precision$_{\text{all}}$	98.57	98.61	98.74	98.50	98.47	98.58	0.11
Recall$_{\text{all}}$	98.77	98.80	98.59	98.76	98.79	98.74	0.09
F1$_{\text{all}}$	98.67	98.71	98.66	98.63	98.63	98.66	0.03
Precision$_{\text{root}}$	96.26	96.33	96.41	96.25	96.05	96.26	0.13
Recall$_{\text{root}}$	96.23	96.40	96.19	96.23	96.06	96.22	0.12
F1$_{\text{root}}$	96.24	96.37	96.30	96.24	96.06	96.24	0.11
Accuracy	97.40	97.45	97.41	97.38	97.38	97.40	0.03
WordAccuracy	90.85	91.01	90.76	90.66	90.84	90.82	0.13
split by roots							
Precision$_{\text{all}}$	95.23	95.06	95.25	95.27	95.92	95.35	0.33
Recall$_{\text{all}}$	95.36	94.75	94.72	95.16	95.22	95.04	0.29
F1$_{\text{all}}$	95.30	94.90	94.98	95.21	95.57	95.19	0.27
Precision$_{\text{root}}$	84.46	80.51	80.38	81.72	85.23	82.46	2.26
Recall$_{\text{root}}$	84.53	80.59	80.44	81.85	85.30	82.54	2.25
F1$_{\text{root}}$	84.50	80.55	80.41	81.78	85.26	82.50	2.25
Accuracy	91.50	90.80	90.92	91.30	91.98	91.30	0.47
WordAccuracy	73.33	69.75	71.14	73.45	75.48	72.63	2.22

Table 10: Segmentation quality of the Morphodict-K dataset using a CNN ensemble

Metric	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**	**StdDev**
random split							
Precision$_{\text{all}}$	98.14	98.24	98.19	98.32	98.39	98.25	0.10
Recall$_{\text{all}}$	98.60	98.55	98.53	98.56	98.50	98.55	0.04
F1$_{\text{all}}$	98.37	98.39	98.36	98.44	98.45	98.40	0.04
Precision$_{\text{root}}$	94.83	95.18	95.04	95.18	95.20	95.08	0.16
Recall$_{\text{root}}$	94.56	94.91	94.83	94.74	94.70	94.75	0.13
F1$_{\text{root}}$	94.69	95.04	94.93	94.96	94.95	94.92	0.13
Accuracy	96.76	96.74	96.85	96.87	96.89	96.82	0.07
WordAccuracy	90.16	90.20	90.07	90.53	90.63	90.32	0.25
split by roots							
Precision$_{\text{all}}$	95.42	95.46	95.28	95.58	95.80	95.51	0.19
Recall$_{\text{all}}$	95.19	95.39	94.98	95.33	95.13	95.21	0.16
F1$_{\text{all}}$	95.31	95.42	95.13	95.46	95.46	95.36	0.14
Precision$_{\text{root}}$	84.55	83.11	84.04	83.47	83.18	83.67	0.62
Recall$_{\text{root}}$	84.35	82.86	84.07	83.32	83.05	83.53	0.65
F1$_{\text{root}}$	84.45	82.99	84.05	83.39	83.11	83.60	0.63
Accuracy	90.82	90.79	90.41	91.00	90.99	90.80	0.24
WordAccuracy	73.37	72.97	72.99	73.85	74.57	73.55	0.67
random split (+lex)							
Precision$_{\text{all}}$	97.77	98.05	97.75	97.88	97.96	97.88	0.13
Recall$_{\text{all}}$	98.34	98.24	98.18	98.48	98.24	98.30	0.12
F1$_{\text{all}}$	98.05	98.15	97.97	98.18	98.10	98.09	0.08
Precision$_{\text{root}}$	93.94	94.42	93.88	94.28	94.15	94.13	0.23
Recall$_{\text{root}}$	93.87	94.24	93.73	93.92	93.62	93.88	0.24
F1$_{\text{root}}$	93.90	94.33	93.81	94.10	93.88	94.01	0.21
Accuracy	96.10	96.35	96.07	96.33	96.20	96.21	0.13
WordAccuracy	88.39	88.97	88.57	89.44	88.92	88.86	0.40
split by roots (+lex)							
Precision$_{\text{all}}$	95.49	95.33	95.51	95.54	95.51	95.48	0.08
Recall$_{\text{all}}$	95.67	95.47	95.47	95.49	95.36	95.49	0.11
F1$_{\text{all}}$	95.58	95.40	95.49	95.51	95.44	95.48	0.07
Precision$_{\text{root}}$	84.63	83.38	84.45	83.47	83.04	83.79	0.70
Recall$_{\text{root}}$	84.29	83.32	84.47	83.34	82.78	83.64	0.71
F1$_{\text{root}}$	84.46	83.35	84.46	83.40	82.91	83.71	0.70
Accuracy	91.45	91.09	90.95	91.11	91.01	91.12	0.20
WordAccuracy	75.64	73.98	74.30	74.67	74.34	74.59	0.64

Table 11: Segmentation quality of the Slounik dataset using a fine-tuned `roberta-small-belarusian model`

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	StdDev
random split							
Precision$_{\text{all}}$	97.69	97.69	97.74	97.78	97.69	97.72	0.04
Recall$_{\text{all}}$	98.22	98.36	98.30	98.22	98.31	98.28	0.06
F1$_{\text{all}}$	97.95	98.02	98.02	98.00	98.00	98.00	0.03
Precision$_{\text{root}}$	93.21	94.08	94.03	93.84	93.42	93.72	0.38
Recall$_{\text{root}}$	92.70	93.77	93.75	93.72	93.10	93.41	0.48
F1$_{\text{root}}$	92.96	93.92	93.89	93.78	93.26	93.56	0.43
Accuracy	95.79	95.92	96.10	96.12	96.01	95.99	0.14
WordAccuracy	87.22	87.52	87.92	88.23	87.73	87.73	0.38
split by roots							
Precision$_{\text{all}}$	95.28	95.20	95.00	95.17	95.79	95.29	0.30
Recall$_{\text{all}}$	96.16	95.72	95.89	95.75	95.91	95.89	0.17
F1$_{\text{all}}$	95.72	95.46	95.45	95.46	95.85	95.59	0.19
Precision$_{\text{root}}$	85.85	83.94	84.78	83.55	84.32	84.49	0.89
Recall$_{\text{root}}$	84.99	83.73	84.14	83.28	84.18	84.06	0.63
F1$_{\text{root}}$	85.42	83.83	84.46	83.42	84.25	84.28	0.75
Accuracy	91.64	90.93	90.74	90.86	91.71	91.17	0.46
WordAccuracy	76.09	73.75	74.03	73.60	76.22	74.74	1.30
random split (+lex)							
Precision$_{\text{all}}$	97.51	97.64	97.70	97.73	97.83	97.68	0.12
Recall$_{\text{all}}$	98.35	98.14	98.20	98.27	98.22	98.24	0.08
F1$_{\text{all}}$	97.93	97.89	97.95	98.00	98.02	97.96	0.05
Precision$_{\text{root}}$	93.47	93.39	93.81	93.58	93.67	93.58	0.17
Recall$_{\text{root}}$	93.06	92.88	93.27	93.12	93.06	93.08	0.14
F1$_{\text{root}}$	93.27	93.13	93.54	93.35	93.37	93.33	0.15
Accuracy	95.80	95.72	96.03	95.96	95.93	95.89	0.13
WordAccuracy	87.67	86.85	87.62	87.88	87.91	87.58	0.43
split by roots (+lex)							
Precision$_{\text{all}}$	95.32	94.99	95.09	95.16	95.73	95.26	0.29
Recall$_{\text{all}}$	96.04	95.73	95.46	95.81	95.60	95.73	0.22
F1$_{\text{all}}$	95.68	95.36	95.27	95.48	95.67	95.49	0.18
Precision$_{\text{root}}$	84.94	82.98	83.44	83.81	84.05	83.84	0.73
Recall$_{\text{root}}$	84.54	82.85	83.26	83.32	83.83	83.56	0.65
F1$_{\text{root}}$	84.74	82.92	83.35	83.56	83.94	83.70	0.69
Accuracy	91.48	90.89	90.63	90.88	91.27	91.03	0.34
WordAccuracy	75.41	73.31	73.65	74.01	75.28	74.33	0.96

Table 12: Segmentation quality of the Slounik dataset using a fine-tuned `SlavicBert` model

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	StdDev
random split							
Precision$_{\text{all}}$	99.27	99.26	99.28	99.25	99.28	99.27	0.01
Recall$_{\text{all}}$	99.54	99.53	99.52	99.53	99.53	99.53	0.01
F1$_{\text{all}}$	99.41	99.40	99.40	99.39	99.41	99.40	0.01
Precision$_{\text{root}}$	96.66	96.64	96.67	96.61	96.62	96.64	0.03
Recall$_{\text{root}}$	96.66	96.62	96.65	96.60	96.60	96.63	0.03
F1$_{\text{root}}$	96.66	96.63	96.66	96.61	96.61	96.63	0.03
Accuracy	98.71	98.68	98.69	98.67	98.70	98.69	0.01
WordAccuracy	95.17	95.14	95.14	95.03	95.13	95.12	0.05
split by roots							
Precision$_{\text{all}}$	96.96	96.99	96.78	96.98	96.87	96.91	0.09
Recall$_{\text{all}}$	96.02	95.43	95.63	95.72	95.87	95.74	0.22
F1$_{\text{all}}$	96.49	96.20	96.20	96.35	96.37	96.32	0.12
Precision$_{\text{root}}$	80.18	78.63	79.49	78.93	79.29	79.31	0.59
Recall$_{\text{root}}$	80.01	78.55	79.50	78.92	79.11	79.22	0.56
F1$_{\text{root}}$	80.10	78.59	79.50	78.92	79.20	79.26	0.57
Accuracy	91.78	91.17	91.35	91.44	91.36	91.42	0.22
WordAccuracy	71.43	69.51	70.01	70.92	70.08	70.39	0.77
random split (+lex)							
Precision$_{\text{all}}$	99.32	99.32	99.33	99.32	99.34	99.33	0.01
Recall$_{\text{all}}$	99.56	99.56	99.54	99.55	99.57	99.55	0.01
F1$_{\text{all}}$	99.44	99.44	99.43	99.43	99.45	99.44	0.01
Precision$_{\text{root}}$	96.79	96.79	96.77	96.77	96.80	96.79	0.01
Recall$_{\text{root}}$	96.79	96.79	96.76	96.76	96.78	96.78	0.01
F1$_{\text{root}}$	96.79	96.79	96.77	96.76	96.79	96.78	0.01
Accuracy	98.76	98.76	98.75	98.75	98.79	98.76	0.02
WordAccuracy	95.36	95.43	95.34	95.35	95.48	95.39	0.06
split by roots (+lex)							
Precision$_{\text{all}}$	97.20	97.03	97.04	97.00	97.03	97.06	0.08
Recall$_{\text{all}}$	96.52	95.92	96.13	96.10	96.21	96.18	0.22
F1$_{\text{all}}$	96.86	96.47	96.58	96.55	96.62	96.62	0.15
Precision$_{\text{root}}$	81.88	79.98	81.16	80.24	80.43	80.74	0.77
Recall$_{\text{root}}$	81.79	79.98	81.11	80.15	80.26	80.66	0.77
F1$_{\text{root}}$	81.83	79.98	81.13	80.20	80.34	80.70	0.77
Accuracy	92.51	91.84	92.25	91.96	91.85	92.08	0.29
WordAccuracy	74.40	71.49	73.04	72.41	71.81	72.63	1.15

Table 13: Segmentation quality of the DeriNet dataset using a fine-tuned `Czert` model

Metric	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average	StdDev
random split							
Precision$_{\text{all}}$	98.72	98.41	98.77	98.41	98.54	98.57	0.17
Recall$_{\text{all}}$	98.76	98.36	98.71	98.57	98.38	98.56	0.18
F1$_{\text{all}}$	98.74	98.38	98.74	98.49	98.46	98.56	0.17
Precision$_{\text{root}}$	95.91	94.91	95.88	95.21	95.31	95.45	0.44
Recall$_{\text{root}}$	95.78	94.47	95.74	95.05	94.87	95.18	0.57
F1$_{\text{root}}$	95.85	94.69	95.81	95.13	95.09	95.31	0.50
Accuracy	97.71	97.05	97.74	97.21	97.25	97.39	0.31
WordAccuracy	92.12	89.97	92.19	90.63	90.52	91.09	1.01
split by roots							
Precision$_{\text{all}}$	93.80	94.80	94.25	95.89	94.60	94.67	0.78
Recall$_{\text{all}}$	96.30	95.68	95.31	93.85	95.44	95.31	0.90
F1$_{\text{all}}$	95.03	95.24	94.78	94.86	95.01	94.98	0.18
Precision$_{\text{root}}$	81.98	81.49	81.67	82.71	81.69	81.91	0.48
Recall$_{\text{root}}$	82.53	82.29	82.21	82.47	81.81	82.26	0.28
F1$_{\text{root}}$	82.25	81.89	81.94	82.59	81.75	82.08	0.34
Accuracy	90.28	90.63	89.90	89.66	90.30	90.15	0.38
WordAccuracy	71.40	72.85	69.78	69.68	71.81	71.10	1.36
random split (+lex)							
Precision$_{\text{all}}$	98.66	98.67	98.67	98.70	98.73	98.69	0.03
Recall$_{\text{all}}$	98.86	98.79	98.90	98.85	98.81	98.84	0.04
F1$_{\text{all}}$	98.76	98.73	98.79	98.77	98.77	98.76	0.02
Precision$_{\text{root}}$	95.95	96.13	96.07	96.18	96.34	96.13	0.14
Recall$_{\text{root}}$	95.90	95.94	96.02	96.08	96.08	96.00	0.08
F1$_{\text{root}}$	95.92	96.03	96.05	96.13	96.21	96.07	0.11
Accuracy	97.75	97.70	97.83	97.82	97.80	97.78	0.05
WordAccuracy	92.31	92.35	92.58	92.64	92.48	92.47	0.14
split by roots (+lex)							
Precision$_{\text{all}}$	96.34	96.37	95.63	96.30	95.83	96.09	0.34
Recall$_{\text{all}}$	95.39	95.17	95.09	94.63	95.13	95.08	0.28
F1$_{\text{all}}$	95.86	95.77	95.36	95.46	95.48	95.59	0.22
Precision$_{\text{root}}$	84.91	84.31	84.78	84.50	83.35	84.37	0.62
Recall$_{\text{root}}$	84.85	84.28	84.75	84.32	83.32	84.30	0.61
F1$_{\text{root}}$	84.88	84.29	84.76	84.41	83.33	84.34	0.61
Accuracy	92.13	91.99	91.32	91.49	91.59	91.70	0.34
WordAccuracy	76.73	76.01	73.15	74.19	74.65	74.95	1.43

Table 14: Segmentation quality of the Morphodict-T dataset using a fine-tuned `RuRoberta-large` model

Metric	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Average**	**StdDev**
random split							
Precision$_{\text{all}}$	99.07	99.06	99.10	99.03	98.97	99.05	0.05
Recall$_{\text{all}}$	99.12	99.16	99.18	99.14	99.11	99.14	0.03
F1$_{\text{all}}$	99.10	99.11	99.14	99.08	99.04	99.09	0.04
Precision$_{\text{root}}$	97.42	97.55	97.65	97.47	97.12	97.44	0.20
Recall$_{\text{root}}$	97.45	97.45	97.56	97.44	97.23	97.43	0.12
F1$_{\text{root}}$	97.43	97.50	97.61	97.45	97.17	97.43	0.16
Accuracy	98.18	98.19	98.25	98.22	98.09	98.19	0.06
WordAccuracy	93.52	93.72	93.76	93.65	93.38	93.61	0.16
split by roots							
Precision$_{\text{all}}$	96.27	95.44	95.50	95.47	96.23	95.78	0.43
Recall$_{\text{all}}$	95.13	94.90	94.24	94.53	94.89	94.74	0.35
F1$_{\text{all}}$	95.69	95.17	94.86	95.00	95.56	95.26	0.36
Precision$_{\text{root}}$	85.01	81.29	79.82	80.92	84.81	82.37	2.38
Recall$_{\text{root}}$	85.15	81.26	79.54	80.82	84.62	82.28	2.47
F1$_{\text{root}}$	85.08	81.28	79.68	80.87	84.71	82.32	2.42
Accuracy	92.15	91.25	90.70	90.96	91.90	91.39	0.62
WordAccuracy	75.66	71.74	70.65	73.03	75.53	73.32	2.24
random split (+lex)							
Precision$_{\text{all}}$	99.02	99.14	99.00	99.00	99.03	99.04	0.06
Recall$_{\text{all}}$	99.20	99.21	99.14	99.18	99.14	99.17	0.03
F1$_{\text{all}}$	99.11	99.17	99.07	99.09	99.09	99.10	0.04
Precision$_{\text{root}}$	97.23	97.53	97.40	97.25	97.41	97.37	0.12
Recall$_{\text{root}}$	97.26	97.46	97.33	97.29	97.39	97.35	0.08
F1$_{\text{root}}$	97.25	97.49	97.37	97.27	97.40	97.36	0.10
Accuracy	98.20	98.29	98.15	98.17	98.15	98.19	0.06
WordAccuracy	93.52	93.80	93.54	93.34	93.51	93.54	0.16
split by roots (+lex)							
Precision$_{\text{all}}$	96.64	95.72	96.29	96.33	96.90	96.38	0.44
Recall$_{\text{all}}$	95.82	95.99	95.37	95.83	96.03	95.81	0.26
F1$_{\text{all}}$	96.23	95.85	95.83	96.08	96.46	96.09	0.27
Precision$_{\text{root}}$	85.74	83.58	82.84	84.60	87.96	84.95	2.01
Recall$_{\text{root}}$	85.64	83.28	82.84	84.53	87.64	84.78	1.93
F1$_{\text{root}}$	85.69	83.43	82.84	84.57	87.80	84.87	1.97
Accuracy	92.91	92.50	92.46	92.77	93.49	92.82	0.42
WordAccuracy	77.95	75.05	75.32	77.85	79.70	77.17	1.96

Table 15: Segmentation quality of the Morphodict-K dataset using a fine-tuned `RuRoberta-large` model