sciendo
DOI 10.2478/jazcas-2021-0051

AN HMM-BASED PoS TAGGER FOR OLD CHURCH SLAVONIC

OLGA LYASHEVSKAYA[1,2] – ILIA AFANASEV[3]
[1] National Research University “Higher School of Economics”, Moscow, Russia
[2] Vinogradov Institute of the Russian Language, Russian Academy of Sciences, Moscow, Russia
[3] Federal State Budgetary Educational Institution of Higher Education “Saint Petersburg State University”, Saint Petersburg, Russia

LYASHEVSKAYA, Olga – AFANASEV, Ilia: An HMM-based PoS tagger for Old Church Slavonic. Journal of Linguistics, 2021, Vol. 72, No 2, pp. 556 – 567.

Abstract: We present a hybrid HMM-based PoS tagger for Old Church Slavonic. The training corpus is a portion of one text, Codex Marianus (40k) annotated with the Universal Dependencies UPOS tags in the UD-PROIEL treebank. We perform a number of experiments in within-domain and out-of-domain settings, in which the remaining part of Codex Marianus serves as a within-domain test set, and Kiev Folia is used as an out-of-domain test set. Analysing by-PoS-class precision and sensitivity in each run, we combine a simple context-free n-gram-based approach and Hidden Markov method (HMM), and added linguistic rules for specific cases such as punctuation and digits. While the model achieves a rather non-impressive accuracy of 81% in in-domain settings, we observe an accuracy of 51% in out-of-domain evaluation, which is comparable to the results of large neural architectures based on pre-trained contextual embeddings.

Keywords: HMM tagger, Old Church Slavonic, PoS tagging, hybrid models, Universal Dependencies

1 INTRODUCTION

Part-of-speech (PoS) tagging has been around for quite a long time as one of the tasks of natural language processing (NLP). Generally, the task is defined as assigning a PoS label to the token, taking into consideration lexical and contextual information. Sometimes, the tags correspond not only to the PoS categories stricto sensu, but also to the morphological features of the token [1], however, we will adhere to the former task definition. The main challenge of PoS tagging is resolving the ambiguity [2]: when considering the token sequence $w_1 \dots w_n$, one should ideally assign one, and only one tag from a tag set $t_1 \dots t_n$ to each token [3].

The methods of PoS tagging for different languages have achieved a certain level of sophistication. Nevertheless, the tagging of less-resourced languages and languages with considerable dialectal and local variation leaves room for experiments. Old Church Slavonic (OCS) is a language preserved in a limited number of manuscripts, mostly ecclesiastical texts copied in monasteries in Croatia, Bulgaria, and Macedonia, that display a mixture of dialectal features. Due to diversity of language material, there is still an open discussion whether all texts belong to the same language. We were inspired by the idea of building a linguistically informed approach to PoS tagging, the results of which would remain stable on a heterogeneous set of texts. The results of computational experiments in this case are to be the subject of interpretation in linguistic terms. Which parts of speech are the hardest ones to tag? What are the particular reasons for this? Why does one method achieve higher efficiency, what makes this particular language work in terms of distribution of tokens? How linguistics may help in selecting more efficient methods of tagging? When one is able to answer these questions, one achieves another aim, the linguistic interpretability of the model.

The main method of PoS tagging presented in the article is the Hidden Markov Model (HMM), enhanced with the Viterbi algorithm ([4], [5]). The task of defining a PoS tag in HMM is reduced to the process of finding the most likely latter HMM state, while taking into consideration all the previous HMM states for all the previous observations [6]. A probable enhancement for the HMM model is a model that defines PoS if it finds one of the two most frequent n-grams that are characteristic for this particular PoS in this particular language. If the PoS is not present in the original training dataset, one might consider the application of some linguistically formed rules, specific for the corpus, used in the testing phase. This article describes different methods of adding an n-gram-based tagger to the model, as well as rules developed for some features that are not present in the training corpus, such as occurrence of fragmentary tokens, punctuation marks, and digits. All experiments are based on the Universal Dependencies (UD) UPOS tagset [7].

2 RELATED WORK

The history of PoS tagging starts with linguistic rule-based systems. This led to years of work of linguists who developed rules for a particular language. With development of technology, scientists started paying attention to a group of statistical methods ([1], [8], [9]). The following step was machine learning methods adapted to the task ([1], [5], [10], [11], [12]). Finally, recent years witnessed the appearance of taggers based on recurrent neural networks [13]. All these methods can be hybridized to form more efficient models ([11], [14], [15]). The most common method by now, though, has been HMM [3], enhanced in different ways, such as the Viterbi algorithm [6], maximum entropy method [3], or transfer learning [15]. Some methods were designed especially for tagging of extremely low-sized datasets, however, the models developed used embeddings pre-trained on bigger datasets that are impossible to get for Old Church Slavonic [16].

PoS tagging of Slavic languages witnessed a specific boom during the former two decades, see, for example, models for Czech ([17], [18]) and Russian ([19], [20], [21]). The recent years, however, sparkled the interest in the older periods of Slavic language history, provoking the appearance of taggers for Slavic languages of earlier periods, including those designed specifically for Old Church Slavonic ([22], [23]).

There are a number of corpus resources that include OCS texts. However, most of them either do not have PoS annotations (as TITUS [24]), or provide restricted access to machine-readable texts. The Manuscript project [25] has a tag set different from the UD tag set. One of the datasets described in the article is based on the UD [26] version of Codex Marianus [27].

Note that various multi-language taggers were designed to work specifically with the UD annotations (e.g., [28], [29]), since the UD repository includes material of typologically different languages annotated under similar schemas. These models have achieved significant success, with approximately a 95% accuracy score when using contextual multilingual embeddings. However, they tend to work really well on homogeneous collections of texts, large collections, and require a lot of space and resources, which make them often environmentally burdening "heavy industrial divisions". Given the fact that OCS is everything but a homogenous and large set of texts [30], they may not perform well enough on it.

3 METHOD AND DATA

For training purposes we use the full text of Codex Marianus, a version tagged for the PROIEL project [27] and then adapted and made available to UD [26]. It consists of ca. 50K tokens split into train, dev, and test parts. The train and dev parts were joined to form a training dataset, the test part was used to test the efficiency of the learning process.

As a baseline model, the n-gram counter was used. This model observed the most frequent n-grams for each PoS on the training dataset, and created a dictionary. We experimented with some enhancements, such as TF-IDF metrics and preliminary decapitalization of tokens.

Another series of training and testing was performed with the HMM model enhanced with the Viterbi algorithm [5]. After a few launches, constantly increasing the amount of data to be fed into the model, accuracy score, the metrics used for measuring the overall model performance, achieved a stable final value of 81%.

RNN-based taggers and basic regression methods were initially considered to be used as well, however, they required either significantly bigger data, or embeddings pre-trained on, once again, significantly bigger data. These models also have a tendency to overfit. This is a crucial fallacy, because the training dataset consists only of one text, Codex Marianus, and other OCS texts may vary greatly from it. The possibility for the model to adapt to new data is of high importance, and this is why these methods were not implemented in the model.

The next stage included different methods of hybridization between HMM and n-gram models, including the regression model, training it to pick from the results of the two models. The regression model used the extra trees method, which previously proved to be efficient during different ML tracks [31]. Of all the different combinations, the hybrid of HMM and 3-gram model, with decapitalization of token in both training and testing phases, and prioritizing of adverb category assignment to HMM, proved to be the most efficient.

The final stage included out-of-domain testing. The text used was Kiev Folia, considered to be the indelible part of the OCS canon, despite having some very specific linguistic features [32].

Kiev Folia has not been tagged previously as a whole text, despite some recent attempts [22]. The original text was taken from the TITUS collection [24] and preprocessed [33]. It contained punctuation marks and digits that were intentionally deleted from ([26], [27]). For recognition of these PoS, the rule-based part of the system was implemented. The model itself is available as an open-source software [34].

4 EXPERIMENTS AND RESULTS

Four series of experiments were conducted. In the first one, two baselines were defined, using the TreeTagger [35] model trained on Bulgarian, the closest relative of OCS. The second one included a series of experiments with n-gram models. The third one presented different attempts at hybridization of HMM and n-gram models. Each of the experiments in these phases was conducted on the Codex Marianus dataset. The fourth series included testing the best model and raw HMM model on the Kiev Folia dataset.

Table 1 shows the mapping between the UD [7] and TreeTagger-Bulgarian [36] PoS tags. The Bulgarian parameters were used, since there are no trained OCS parameters for TreeTagger, and its source code, essential for the training process, is closed. The by-tag performance of the baseline model is presented in table 2. These results are the bare minimum that any model trained for OCS should beat. The baseline results are acquired via the applying of a loosely similar model, trained on a loosely similar language, mainly connected with OCS genetically, and not typologically. We also applied Russian and Slovak training parameters, however, accuracy with these achieved only 26% and 1% respectively, due to crucial differences in the tagsets.

UD	Explication	TreeTagger
ADJ	adjective	A, Mo, Md, My, H
ADV	adverb	D
INTJ	interjection	I
NOUN	noun	Nc
PROPN	proper noun	Np
VERB	verb	Vn, Vp
ADP	adposition (preposition)	R
AUX	auxiliary	Vx, Vy, Vi
CCONJ	coordinate conjunction	Cc, Cr, Cp
DET	determiner (adj. pronoun)	Ps
NUM	numeral	Mc
PART	particle	T
PRON	pronoun	Pp, Pd, Pr, Pc, Pi, Pf, Pn
SCONJ	subordinate conjunction	Cs
X	non-word	(if not tagged)

Tab. 1. Mapping UD tags onto the TreeTagger-Bulgarian

The aim of the next experiment series was to train a model that chooses the most frequent n-grams for each PoS (in the case of OCS, the two most frequent n-grams for each PoS are most helpful). During the prediction phase, the model tries to find each of these n-grams in the token, and, if not found, assigns verb, as it is the most common tag in both the training and test dataset (8356 and 2281 tokens respectively, 10637 overall). The results for different n (2, 3, 4) are presented in table 2. The 3-gram model proved to be the most efficient, though it barely outperformed the baseline, and needed further enhancements.

PoS	TreeTagger	2-grams	3-grams	4-grams
VERB	78.65	43.93	96.02	**98.88**
AUX	1.85	-	77.85	**96.04**
ADV	2.41	**70.16**	67.08	67.08
NOUN	**30.49**	16.4	-	-
PRON	4.34	66.09	83.92	**87.83**
CCONJ	36.92	**45.91**	20.66	20.66
ADP	25.97	**53.35**	49.48	49.59
ADJ	25.67	51.64	**98.8**	63.49
INTJ	8.47	**19.38**	18.12	18.12
SCONJ	2.17	31.66	**51.55**	48.97
DET	3.73	18.68	**43.9**	-
PROPN	-	**5.26**	-	-
NUM	-	-	-	-
X	-	19.66	**20.82**	20.26
Total	31.54	30.25	**32.19**	31.43

Tab. 2. Accuracy score for PoS tagging with TreeTagger-bg, 2-gram, 3-gram, and 4-gram models. The best results in comparison, here and then, are given in **bold**

N-grams were coming mostly from the first n characters of tokens of the particular PoS. The average distance from the first and the last character of the token to the first character of n-gram is provided in table 3.

PoS	Distance from the beginning	Distance from the end
VERB	0	8.1
ADJ	0.29	6.91
X	0	6
Average	0.07	7.79

Tab. 3. Average n-gram distribution for selected PoS

Then, the possibility of some enhancements to the 3-gram model was considered. For instance, we implemented a rule-based system that normalizes words that are abbreviated and covered by *titlo* (a tilde-like character) in the original texts. However, just deleting *titlo* proved to be more useful, since contractions such as *ic* (which is for the most frequent proper noun in the corpus, being *icoycъ* 'Jesus'), are more recognizable by the model.

Slight improvements of the n-gram model quality were made with decapitalization of all the tokens in the dataset. The special symbol '#' was added to the start and the end of each token, which enhanced efficiency by more than 7 per cent overall.

After that, further attempts were made, using the length of token as a criterion, counting the digraph *oy* [u] as a single vowel and ignoring repeating symbols, and introducing the TF-IDF weighting for n-grams count. It seems that token length was too ambiguous, digraphs and repeating symbols were too rare to actually influence the *status quo* for two of the most frequent n-grams, and the TF-IDF weighting did not actually make a difference for the n-grams that often. The results of the experiments are provided in the table 4.

PoS	3-gram	3 + D	3 + DB	3 + DBT	3 + DBL	3 + DBR
VERB	96.02	**96.13**	54.96	54.96	51.84	54.96
AUX	77.85	77.85	**100**	96.3	**100**	**100**
ADV	67.08	**67.39**	61.86	61.86	16.97	61.86
NOUN	-	-	**37.99**	**37.99**	36.79	**37.99**
PRON	83.92	83.92	**89.22**	**89.22**	-	**89.22**
CCONJ	20.66	21.09	**85.16**	**85.16**	46.95	**85.16**
ADP	49.48	**49.49**	40.49	40.49	-	40.49
ADJ	98.8	**98.8**	86.02	86.02	88.51	86.02
INTJ	18.12	18.57	**20.16**	**20.16**	-	**20.16**
SCONJ	51.55	51.55	78.13	78.13	**80**	78.13
DET	**43.9**	**43.9**	18.75	18.75	26.55	18.75
PROPN	-	-	93.81	93.81	**94.64**	93.81

PoS	3-gram	3 + D	3 + DB	3 + DBT	3 + DBL	3 + DBR
NUM	-	-	50	**56.9**	45	50
X	20.82	20.83	23.74	23.67	**28.01**	23.74
Total	32.19	32.45	**39.17**	39.15	38.27	**39.17**

Tab. 4. Accuracy score for PoS tagging with 3-gram model (3-gram), enhanced with decapitalization (3 + D), decapitalization, and marks of token borders (3 + DB), the latter two and TF-IDF weighting (3 + DBT), or token length counter (3 + DBL), or scoring repeating symbols, and digraphs as one symbol (3 + DBR)

These results were still very low, so the next decision was to train a raw HMM model. The results are presented in table 5.

PoS	HMM
VERB	68.79
AUX	98.77
ADV	60.31
NOUN	99.04
PRON	86.51
CCONJ	99.65
ADP	98.7
ADJ	56.36
INTJ	91.53
SCONJ	70
DET	73.13
PROPN	87.33
NUM	92.56
X	20
Total	81.04

Tab. 5. Accuracy score for PoS tagging with raw HMM with Viterbi algorithm (HMM) model on the UD OCS test dataset module

As can be seen, both HMM and n-gram models tend to be biased towards a particular PoS. HMM is better at detecting nouns, though sometimes it fails in distinguishing them from other PoS. The same may be said of the n-gram model and verb. N-gram may be of better utility, while searching for particular PoS, like X, verb, and adjective. And, simultaneously, it may have a negative effect on the overall quality of tagging. So we built a set of hybrid HMM and n-gram models following the path paved by TreeTagger [35], but making them more adapted to the structure of the OCS data. The hybrids employed the following scheme. After the HMM model made preliminary tagging, the 3-gram model, with or without enhancements, checked tokens once again, assigning them to a preliminary defined PoS. These were adjective, adverb, verb, and X (non-word), for which the n-gram model chose the correct tag with a higher probability than the HMM model. However, the first test defined that adverb tagging with the 3-gram model decreases overall efficiency, and the accuracy of the noun tag prediction decreases anyway. However, the 3-gram model additionally tags only adjective, verb, and X, total accuracy increases, at the cost of noun accuracy. A slight increase in efficiency, achieved by decapitalization, remained. In contrast, explicit representation of the token borders made a slight decrease in performance.

Almost each hybrid significantly increased accuracy of X detection. This is due to the fact that there is a very small number of X in the test dataset, and the difference is not more than 5 correctly assigned tags.

The final experiment included a regression model that learned to predict the correct tag on the basis of HMM and 3-gram prediction. Accuracy of this model is slightly unstable, due to the rounding implementation in Python. Having said that, it was still the baseline model which produced better results.

The results of experiments with hybrid models are given in table 6.

PoS	Baseline	HMM + 3	HMM + 3 – ADV	HMM + 3 + D – ADV	HMM + 3 + DB – ADV	HMM + 3 + DB – ADV + ETR
VERB	68.79	71.37	72.69	**72.82**	55.46	72.69
AUX	**98.77**	**98.77**	**98.77**	**98.77**	**98.77**	**98.77**
ADV	60.31	**80.06**	60.31	60.31	60.12	57.42
NOUN	**99.04**	94.66	98.08	98.08	96.16	98.2
PRON	**86.51**	72.79	**86.51**	**86.51**	86.13	73.32
CCONJ	**99.65**	**99.65**	**99.65**	**99.65**	**99.65**	6.83
ADP	**98.7**	98.18	98.18	98.18	82.47	**98.7**
ADJ	56.36	52.94	**56.9**	**56.9**	55.08	56.36
INTJ	**91.53**	**91.53**	**91.53**	**91.53**	**91.53**	**91.53**
SCONJ	**70**	63.04	**70**	**70**	**70**	19.57
DET	**73.13**	**73.13**	**73.13**	**73.13**	64.93	25.37
PROPN	**87.33**	86.3	**87.33**	**87.33**	83.22	87.33
NUM	**92.56**	91.74	91.74	91.74	90.08	63.64
X	20	**60**	**60**	**60**	**60**	20
Total	81.04	80.6	81.79	**81.82**	75.85	69.62

Tab. 6. Accuracy score for PoS tagging with HMM model (HMM), enhanced with 3-gram model (1), 3-gram model that does not make additional predictions for adverbs (2), the latter with decapitalization (3), and with explicit statement of token beginning and ending (4), the latter and the Extra Trees Regressor, picking the best possible option

For the following out-of-domain experiment run on the Kiev Folia dataset, architecture HMM + 3 + D – ADV was taken, because it demonstrated the best results in within-domain settings. Basic HMM model performed as a baseline in this case. The HMM + 3 + D – ADV model was additionally enhanced with rules that help to define punctuation marks, digits, and fragmentary tokens. Enhanced HMM model performed better (50.93%) than the baseline one (32.64%) on Kiev Folia, as it had done previously on Codex Marianus. Quite expectedly, both models make significantly more mistakes than they did on the UD OCS dataset. Examples of Kiev Folia tokens tagging are provided in table 7.

Token	HMM tag	Enhanced model tag	Correct tag
Твоемоу 'yours-DAT/LOC'	NOUN	ADJ	ADJ
~*ИВ*~ '12'	NOUN	DIGIT	DIGIT
. '.'	NOUN	PUNCT	PUNCT
приведетъ 'lead-FUT(I)'	NOUN	VERB	VERB
присно 'always'	NOUN	VERB	ADV
приснодѣвѣ 'Mary, mother of Jesus-DAT/LOC'	NOUN	VERB	NOUN

Tab. 7. Examples of Kiev Folia dataset token tagging

5 ANALYSIS AND DISCUSSION

As one can see, the best models in our experiments achieve a rather non-impressive accuracy of 81% in in-domain settings and an accuracy of 51% in out-of-domain evaluation. In comparison, the UDpipe 2 neural tagger that employs the character level and multilingual BERT embeddings [29] achieves and accuracy of 97% and 50% respectively, being exposed to a larger amount of training data. The main explanation for such a dramatic drop is different letter distribution, different PoS tags distribution and even new letters and punctuation marks that have not been seen in training. Even the most frequent token, the coordinate conjunction 'and', is mostly presented as *и* in Codex Marianus and as *i* in Kiev Folia. We do not suggest the rule-based component of the tagger to be too specifically tuned to one particular text. Rather, the rules cover the PoS tags that are (almost) missing in the training set. The other part of work is done by the n-gram-based add-on that is based on assumption that there are morphemes, subtokens or other stable character combinations that can serve as a cue to the PoS identification. We believe that there is a space for future improvements of probabilistic models based on attention to most frequent character n-grams.

The analysis of the PoS confusion matrix on the in-domain and out-of-domain shows that verbs are frequently tagged as nouns and vice versa; adjectives with one-character endings are incorrectly labeled nouns (*петров-ъ* '*of Peter.POSS*', *мьног-ы* 'many'). At the same time, the closed-class PoS tags are identified mostly correctly in the in-domain test set, the only source of errors being the homonymy of prepositions and adverbs and conjunctions and adverbs or pronouns. In the out-of-domain test set, a lot of words from the closed-class PoS are erroneously tagged as nouns.

The disadvantage of the method is a partial loss of the context sensitivity. Thus, the noun *весь* ‘village’ is labeled determinant, as it has a homonymous and much more frequent reading ‘all’. Analogically, единъ ‘one’ is labeled numeral when it refers to indefinite pronoun. Another known issue is nominalizations and other same-root and same-prefix words that get the tag of the most frequent word in their word formation family (usually, the basic word of the family). As a result, nouns that include n-gram *г҃л* in the root (*г҃лъ* ‘verb’, *г҃лсъ* ‘voice’) and nouns with the prefix *при* (*притъчи* ‘parables’, *пришельца* ‘newcomer’) are incorrectly labeled verb. Apparently there should be found a sensible trade-off between the context sensitivity and subtoken recognition.

6 CONCLUSION

An HMM-based PoS tagger for OCS was developed, with some enhancements for both overall performance (n-gram models), and performance on specific PoS, such as digits and punctuation marks. The model achieves the accuracy score of 81% on the UD OCS dataset, and 51% on Kiev Folia dataset, which may satisfy the criterion of model scalability to out-of-domain data. HMM model, despite being in use since the early 1990s, was demonstrated to be still useful for a specific case of heterogeneous train and test data. What is more, the model seems to be operating better than pre-trained TreeTagger, that it was inspired by, and RFTagger [37] (14% overall accuracy on Kiev Folia).

The model is less accurate on UD OCS dataset than the UD multilingual models [29], however, it proves itself to be more useful for Kiev Folia dataset, due to the implementation of the rule-based systems. The results of out-of-domain evaluation yet again raise the question of how linguistically heterogeneous the OCS canon actually is. Apart from being practically useful for the OCS data annotation tasks, the cross-variant PoS tagging can provide actual insights into the scale of their difference.

The heterogeneity data case was a new challenge for OCS PoS tagging (comparing to [22] and [35]). And, with more texts being translated into a machine-readable form, and this model achieving 51% accuracy (which, as we hope, is not the best results that might be achieved), this challenge is to be faced in the future. Probably the following enhancements, like the ones that were conducted, are going to improve the overall results. The main aim here is to improve scalability of the model, not its efficiency for a single dataset.

References

[1] Behera, P. (2017). An Experiment with the CRF++ Parts of Speech (POS) Tagger for Odia Language in India, 17(1), pages 18–40.

[2] Loftsson, H. (2008). Tagging Icelandic text: A linguistic rule-based approach. Nordic Journal of Linguistics, 31(1), pages 47–72.

[3] Dandapat, S., Sarkar, S., and Basu, A. (2007). Automatic Part-of-Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 221–224, Association for Computational Linguistics.
[4] Rajendran, S., and Krishnakumar. K. (2019). A Comprehensive Study of Shallow Parsing and Machine Translation in Malaylam. Coimbatore: Amrita Vishwa Vidyapeetham, 295 p.
[5] Uludoğan, G. (2018). HMM POS tagger. Accessible at: https://github.com/gokceuludogan/hmm-pos-tagger.
[6] Jurish, B. (2003). A Hybrid Approach to Part-of-Speech Tagging. Berlin: Berlin-Brandenburgishe Akademie der Wissenschaften, 2003, 27 p.
[7] (UD) UPOS tag set. Accessible at: https://universaldependencies.org/u/pos/.
[8] Mohamed Elhadj, Y. O. (2009). Statistical Part-of-Speech Tagger for Traditional Arabic Texts. Journal of Computer Science, 5(11), pages 794–800.
[9] Danso, S., and Lamb, W. (2014). Developing an Automatic Part-of-Speech Tagger for Scottish Gaelic. In Proceedings of the First Celtic Language Technology Workshop, pages 1–5, ACL.
[10] Mirzanezhad, Z., and Feizi-Derakhshi, M.-R. (2016). Using morphological analyzer to statistical POS Tagging on Persian Text. IJCSIS, 14(8), pages 1093–1103.
[11] Abumalloh, R. A., Al-Sarhan, H. M., Ibrahim, O. B., and Abu-Ulbeh, W. (2016). Arabic Part-of-Speech Tagging. Journal of Soft Computing and Decision Support Systems, 3(2), pages 45–52.
[12] Kumar, S. S., Kumar, M. A., and Soman, K. P. (2016). Experimental analysis of Malayalam PoS tagger using EPIC framework in Scala. ARPN Journal of Engineering and Applied Sciences, 11(13), pages 8017–8023.
[13] Gambäck, B., Olsson, F., Argaw, A. A., and Asker, L. (2009). Methods for Amharic Part-of-Speech Tagging. In Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages, pages 104–111, ACL.
[14] Saharia, N., Das, D., Sharma, S., and Kalita, J. (2009). Part of Speech Tagger for Assamese Text. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 33–36, World Scientific Publishing Co Pte Ltd.
[15] Reddy, S., and Sharoff, S. (2011). Cross Language POS Taggers (and other Tools) for Indian Languages: An Experiment with Kannada using Telugu Resources. In Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 11–19, Asian Federation of Natural Language Processing.
[16] Kann, K. et al. (2018). Character-level supervision for low-resource POS tagging. In Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP, pages 1–11, Association for Computational Linguistics.
[17] Straka, M., Strakova, J., and Hajič, J. (2019). Czech Text Processing with Contextual Embeddings: POS Tagging, Lemmatization, Parsing and NER. In Proceedings of 22nd International Conference "Text, Speech and Dialogue" 2019, pages 137–150, TSD.
[18] Hajič, J., and Hladká, B. (1998). Czech language processing, POS tagging. Accessible at: https://ufal.mff.cuni.cz/czech-tagging/HajicHladkaLREC1998.pdf.
[19] Korobov, M. (2015). Morphological Analyzer and Generator for Russian and Ukrainian Languages. Analysis of Images, Social Networks and Texts, pages 320–332.
[20] Bird, S., Loper, E., and Klein, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc, 479 p.

[21] Segalovich, I. (2003). A Fast Morphological Algorithm with Unknown Word Guessing Induced by a Dictionary for a Web Search Engine. In Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications, pages 273–280, MLMTA'03, June 23–26, 2003, Las Vegas, Nevada, USA.
[22] Eckhoff, H. M., and Berdicevskis, A. (2015). Linguistics vs. digital editions: The Tromsø Old Russian and OCS Treebank. Scripta & e-Scripta 2015, 14(15), pages 9–25.
[23] Pedrazzini, N., and Eckhoff, H. M. (2021). OldSlavNet: A scalable Early Slavic dependency parser trained on modern language data. Software Impacts, 8. Accessible at: https://www.sciencedirect.com/science/article/pii/S2665963821000117.
[24] TITUS. Accessible at: http://titus.uni-frankfurt.de/indexe.htm.
[25] Manuscript. Accessible at: http://manuscripts.ru/.
[26] Zeman, D., Nivre, J., Abrams, M. et al. (2020). Universal Dependencies 2.7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. Accessible at: http://hdl.handle.net/11234/1-3424.
[27] Haug, D. T. T., and Jøhndal, M. L. (2008). Creating a Parallel Treebank of the Old Indo-European Bible Translations. In Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008), pages 27–34, ACM, New York, NY.
[28] Straka, M., Hajič, J., and Straková, J. (2016). UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia. Accessible at: https://ufal.mff.cuni.cz/~straka/papers/2016-lrec_udpipe.pdf.
[29] Straka, M., Straková, J., and Hajič, J. (2019). Evaluating Contextualized Embeddings on 54 Languages in POS Tagging, Lemmatization and Dependency Parsing. In ArXiv.org Computing Research Repository, ISSN 2331-8422, 1904.02099.
[30] Kamphuis, J. (2020). Verbal Aspect in Old Church Slavonic: A Corpus-based Approach. Leiden: Brill, 329 p.
[31] Strobl, C., Malley, J., and Tutz, G. (2009). An Introduction to Recursive Partitioning: Rationale, Application and Characteristics of Classification and Regression Trees, Bagging and Random Forests. Psychological Methods, 14(4), pages 323–348.
[32] Kiev Folia. Accessible at: http://www.schaeken.nl/lu/research/online/editions/kievfol.html.
[33] Afanasev, I. (2020). Korpus staroslavianskogo iazyka: nedostaiushchee zveno v diakhronicheskoi slavistike. In Slavica iuvenum XXI: sbornik trudov mezhdunarodnoi nauchnoi konferentsii Slavica iuvenum 2020, March 31–April 1, 2020, pages 13–21, Ostravskii universitet, Ostrava.
[34] Project GitHub Repository. Accessible at: https://github.com/The-One-Who-Speaks-and-Depicts/hmm-pos-tagger.
[35] Helmut, S. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In Proceedings of International Conference on New Methods in Language Processing, pages 1–9, Manchester, UK. Accessible at: https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger2.pdf.
[36] Simov, K., Osenova, P., and Slavcheva, M. (2004). BTB-TR03: BulTreeBank Morphosyntactic Tagset. Accessible at: http://bultreebank.org/wp-content/uploads/2017/06/BTB-TR03.pdf.
[37] Schmid, H., and Laws, F. (2008). Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging. Accessible at: https://cis.lmu.de/~schmid/papers/Schmid-Laws.pdf.