Has been issued since 2002
https://mongoloved.kigiran.com/jour

Vol. 17, Is. 2, pp. 371–390, 2025
DOI: 10.22162/2500-1523-2025-2-371-390

**МОНГОЛОВЕДЕНИЕ**
*(Монгол судлал)*
*(Mongolian Studies)*
ISSN 2500-1523 (Print)
ISSN 2712-8059 (Online)

**ЯЗЫКОЗНАНИЕ**



UDC 811.512.3

# Neural Network Models of a Grammar Parser for the Kalmyk Language: Training Experience

*Abina D. Kukanova*[1],
*Viktoria V. Kukanova*[2]

[1] Kalmyk Scientific Center of the RAS (8, Ilishkin St., 358000 Elista, Russian Federation)
Junior Research Associate

0009-0001-8103-7504. E-mail: kukanovaabina[at]gmail.com

[2] Kalmyk Scientific Center of the RAS (8, Ilishkin St., 358000 Elista, Russian Federation)
Cand. Sc. (Philology), Director, Senior Research Associate

0000-0002-7696-4151. E-mail: kukanovavv[at]kigiran.com



**Abstract.** *Introduction.* Kalmyk language presents unique challenges for NLP due to its agglutinative rich morphology and limited available resources. The *objective* is to consider various neural network models of grammar analysis for the Kalmyk language. *Materials and Methods.* Several neural network models were selected for training: Lemma Accuracy, Levenshtein Lemma Distance, Morph Accuracy, Morph F1. Neural network model training methods, analysis, and comparison methods were used. The training dataset used consisted of an organizational part in depth of 2 495 sentences (including 35 049 tokens), a validation part in depth of 311 sentences (including 3 991 tokens),

УДК 811.512.3

# Нейросетевые модели грамматического анализатора для калмыцкого языка: опыт обучения

*Абина Денисовна Куканова*[1],
*Виктория Васильевна Куканова*[2]

[1] Калмыцкий научный центр РАН (д. 8, ул. им. И. К. Илишкина, 358000 Элиста, Российская Федерация)
младший научный сотрудник

0009-0001-8103-7504. E-mail: kukanovaabina[at]gmail.com

[2] Калмыцкий научный центр РАН (д. 8, ул. им. И. К. Илишкина, 358000 Элиста, Российская Федерация)
кандидат филологических наук, директор, старший научный сотрудник

0000-0002-7696-4151. E-mail: kukanovavv[at]kigiran.com



**Аннотация.** *Введение.* Калмыцкий язык представляет особые трудности для обработки естественного языка из-за своей богатой агглютинативной морфологии и ограниченных доступных ресурсов. *Цель* — проанализировать различные нейросетевые модели грамматического анализатора для калмыцкого языка. *Материалы и методы.* Для обучения выбраны несколько нейросетевых моделей: Lemma Accu-racy, Lemma Le-venshtein Distance, Morph Accuracy, Morph F1. Применялись методы обучения нейросетевых моделей, методы анализа, сравнения. Для обучения использовался датасет, состоящий из тренировочной части в

and a test part in depth of 313 sentences (including 3 627 tokens). *Results.* This paper proposes a high-performing morphological analyzer for Kalmyk language using neural network techniques. The analyzer is able to jointly predict a lemmata and morphological tags for each word in a sentence. Due to the scarcity of the data, morphological analyzers for low-resource languages often utilizes rule-based and statistical approaches. However, there are few studies based on deep learning approaches. Firstly, our model inputs word embedding based on characters and contextual embeddings generated by the pretrained cross lingual model XLM-RoBERTa. Secondly, the proposed model is based on a sequential architecture which inputs surface words and predicts minimum edit actions between surface words and lemmas instead of predicting characters in lemmas. Thirdly, our system does not require pretrained embeddings for the Kalmyk language and additional morphological segmentation tools. We conducted several experiments to show that our model outperforms other models.

**Keywords:** Kalmyk language, morphological analyzer, parts of speech, grammemes, neural network models, NLP, XLM-RoBERTa

**Acknowledgments.** The reported study was funded by Russian Science Foundation, project number 25-78-20008 “Developing Research Tools and Conducting Comprehensive Studies of the Mongolic Languages and Their Languages: Applying Big Data Tools for the Analysis of Dictionaries and Corpora”.

**For citation:** Kukanova A. D., Kukanova V. V. Neural network models of grammar parser for the Kalmyk language: training

объеме 2 495 предложений (в их числе 35 049 токенов), валидационной части в объеме 311 предложений (в их числе 3 991 токена), тестовой части в объеме 313 предложений (в их числе 3 627 токенов). *Результаты.* В данной статье предлагается высокопроизводительный морфологический анализатор для калмыцкого языка, использующий методы нейронных сетей. Анализатор способен совместно предсказывать леммы и морфологические теги для каждого слова в предложении. Из-за нехватки данных морфологические анализаторы для языков с низкими ресурсами часто используют основанные на правилах и статистические подходы. Однако существует мало исследований, основанных на подходах глубокого обучения. Во-первых, наша модель использует вложения слов на основе символов и контекстуальных вложений, сгенерированных предобученной кросс-языковой моделью XLM-RoBERTa. Во-вторых, предлагаемая модель основана на последовательной архитектуре, которая вводит поверхностные слова и предсказывает минимальные действия редактирования между поверхностными словами и леммами вместо предсказания символов в леммах. В-третьих, наша система не требует предобученных вложений для калмыцкого языка и дополнительных инструментов морфологической сегментации. Мы провели несколько экспериментов, чтобы показать, что наша модель превосходит другие модели.

**Ключевые слова:** калмыцкий язык, морфологический анализатор, части речи, граммемы, нейросетевые модели, NLP, обучениеб XLM-RoBERTa

**Благодарность.** Исследование проведено при финансовой поддержке РНФ в рамках проекта «Разработка инструментария и комплексные исследования монгольских языков и их диалектов (с применением технологий анализа больших массивов данных словарных и корпусных материалов (№ 25-78-20008).

**Для цитирования:** Куканова А. Д., Куканова В. В. Нейросетевые модели грамматического анализатора для калмыцкого

experience. *Mongolian Studies (Elista)*. 2025. Vol. 17. Is. 2. Pp. 371–390. (In Russ.). DOI: 10.22162/2500-1523-2025-2-371-390

языка: опыт обучения // Монголоведение. 2025. Т. 17. № 2. С. 371–390. DOI: 10.22162/2500-1523-2025-2-371-390

## 1. Introduction

Morphological analysis, a fundamental step of processing the text in many NLP pipelines, deals with the structure of words by identifying the constituent morphemes. It usually consists of two tasks: lemmatization and morphological tagging [Baxi, Bhatt 2024]. Lemmatization is the process of reducing a word to its base or root form [Kote et al. 2019]. Lemmatization is crucial for languages with rich morphology since each word can have several inflectional variants, which increases the vocabulary size and a number of out-of-vocabulary (OOV) words. Morphological tagging is a subtask in which morphosyntactic information and a part of speech are assigned to each word [Heigold et al. 2017]. Compared to part-of-speech tagging, morphological tagging not only adds a part of speech, but also other grammatical features like case, gender, number, tense, etc. Morphological analyzers are essential tools for many NLP applications like machine translation, language modeling, syntactic parsing, information retrieval (IR), spell checkers and named entity recognition (NER). Thus, the development of an effective morphological analyzer has a greater impact on the computational recognition of a language.

For the low-resource languages, building morphological analyzers presents a unique set of challenges due to scarcity of the resources since it requires thousands of annotated sentences to train. Therefore, the lack of training data can be a serious obstacle, hindering the progress of downstream NLP applications and, ultimately, impacting the digital vitality of endangered languages. Many people doubt the importance of preserving low-resource languages without thinking about the consequences. The extinction of a language represents an irreparable loss of cultural heritage, scientific knowledge and important aspects of human history. Oral traditions, literature, and historical documents will be lost and forgotten. The very existence of digital resources like morphological analyzers can contribute significantly to protection and revival of endangered languages. Thus, the development of a morphological analyzer acquires additional importance, becoming not just a technical challenge, but a crucial step in supporting language preservation and cultural heritage.

Kalmyk is a member of the Mongolian language family, natively spoken in the Kalmykia Republic in the west of the Russian Federation, and in parts of western China and western Mongolia. Along with Russian, the Turkic, Ugric and Tungusic elements that joined the Kalmyk ethnic group had a certain influence on the Kalmyk language. According to UNESCO, Kalmyk is “definitely endangered”. As an agglutinative language, word formation occurs by sequentially attaching to the root. Each affix in a derived word retains its independence and has only its inherent meaning. However, prefixes are not common in Kalmyk, and affixes are usually added after a root or at the end of a sentence.

Morphological analyzers for low-resource agglutinative language are often based on rule-based and statistical approaches due to scarcity of the resources, and there are just few studies that focus on deep learning approaches. Rule-based and machine learning based approaches require deep linguistic expertise, rely on heavy feature engineering, and ignore local context. Also, there are no available pretrained

embeddings or morphological tools for Kalmyk which makes this task even more difficult. Therefore, we propose a joint morphological analyzer, which does not require the above mentioned tools. Firstly, our model's input embedding layer consists of three parts: character-level embedding, word embedding and contextual embeddings generated by using the pretrained cross lingual model XLM-RoBERTa. We implemented this method from [Abudouwaili et al. 2023]. Secondly, our model is based on a sequential architecture which inputs surface words and predicts minimum edit actions between surface words and lemmas instead of predicting characters in lemmas [Yildiz, Tantuğ 2019]. We trained several models for Kalmyk to compare and analyze our results with theirs.

The structure of this paper is the following: in 'Morphological Tools Overview' we investigate various approaches of developing morphological analyzers, the section 'Data' describes the structure and annotation of Kalmyk dataset, the section 'Methods' presents the architecture of our model, in 'Experimental results' we run several experiments to evaluate our analysis, the section 'Conclusion' summarizes our work.

## 2. Morphological Tools Overview

Morphological analyzer is a crucial component in numerous NLP applications. Researchers have developed a diverse amount of morphological analyzers for various languages, many of which are publicly accessible. This section provides an overview of the popular methodologies employed in building these tools, while noting their advantages and disadvantages. Our exploration will cover:

1. Rule-based methods, which rely on handcrafted linguistic rules.
2. Machine learning based approaches, which learn probabilistic relationships from pre-classified data.
3. Deep learning based methods, which focus on building multi-layered neural networks.

### *2.1. Rule based approaches*

Rule based systems have been developed since the earliest times. These methods are based on a set of predefined linguistic rules and patterns developed by experts. Linguists formulate these rules, which can encompass such aspects as phonology, inflectional and derivational grammar, syntax, often utilizing regular expressions to define word formation patterns. The developed rules are then applied to the text to capture matched patterns. During analysis, the created set of rules is constantly being modified and updated to improve accuracy and performance. This approach is easily interpretable since rules are explicitly defined. Rule based systems show good results if the language is highly structured and unambiguous.

Two-level morphology is the first model in the history of computational linguistics designed to analyze and generate morphologically complex languages. K. Koskenniemi [Koskenniemi 1996] proposed this model in his dissertation, which is based on 3 fundamental ideas:

1. The rules are character-to-character constraints applied in parallel rather than sequentially, like substitution rules.
2. Constraints can relate to the lexical context, the surface context, or both contexts at the same time.
3. Lexical lookup and morphological analysis are performed together.

This method represents words on two levels: lexical and surface levels. Rules in this system describe how the lexical representation of a word is transformed into its surface form, taking into account various morphological processes. In generative morphology, the rules can only produce word forms, while in a two-level morphology model, the rules can generate and detect word forms.

Two-level morphology is implemented using a set of finite state transducers (FSTs), that are responsible for a certain type of correspondence between the lexical and the surface level. The machine moves between the states based on the input symbol, while it outputs the corresponding output symbol.

One of the most popular methods for the implementation of FST based is Helsinki Finite State Toolkit (HFST) developed by [Lindén et al. 2009]. HFST, an open-source software for the developing and application of FST and finite automata (FSAs), allows users to build morphological analyzers and generators. The tool was applied to numerous languages like Wolaytta [Gebreselassie et al. 2018], Evenki [Zueva et al. 2020].

Paradigm based approach is another rule-based method to build morphological analyzers. Its core idea is to map an input word to its corresponding paradigm. A paradigm defines all possible word forms for a given stem and set of grammatical features. It may require handling ambiguity if the same rule may be present in one paradigm or a word fits multiple paradigms. J. Baxi, P. Patel, B. Bhatt [Baxi et al. 2015] developed a morphological analyzer for Gujarati language combining s statistical, knowledge based and paradigm based approaches.

Another rule-based method is based on stemming. It employs a stemmer with rules, often using suffix stripping. Stemming based approach is simple and doesn't require a lookup table, but not sufficient for lemmatization or morphological tagging. Also, the linguistic rules have to be accurately specified.

*2.2. Machine learning based approaches*

Rule-based methods were used in many NLP applications like machine translation, named entity recognition and information retrieval but over time researchers realized that this approach suffers from several major limitations. One primary drawback is the considerable time and resource investment required for the development and maintenance. Creating rule-based systems necessitates manual creation of linguistic rules describing language's inflectional patterns, derivational processes and numerous exceptions. This approach not only demands deep linguistic expertise but also becomes laborious as the complexity of the language increases. Among other things, rule-based systems lack flexibility. As languages evolve and new words or structures emerge, they struggle with OOV words. This inflexibility makes it complicated to adapt to the nuances and variability of natural language. What is also important to note is that rule-based systems struggle with disambiguation in the word formation rules. The same morpheme can have multiple meanings or functions, and rule-based systems face difficulties to uniquely identify a part of speech.

With the above-mentioned disadvantages and the growing popularity of machine learning based methods, NLP researchers have moved their focus from traditional rule-based systems to machine learning based methods. The idea of this approach is that a statistical model learns probabilistic relationships between morphemes and their

corresponding tags from annotated data. Researchers trained various supervised and unsupervised models to solve this problem.

Unsupervised morphological analyzer allows to identify morphemes and their functions within words algorithmically, without any supervision. Z. S. Harris [Harris 1955] introduced Letter Successor Variety models (LSV) that are based on the analysis of the distribution of letter sequences in words. These models try to find morpheme boundaries based on changes in this distribution. A high variety of letter sequences may indicate a morpheme boundary.

Another method used to develop a morphological analyzer was Minimum Description Length (MDL). This method seeks to find the most compact representation of the data. In the context of morphology, they are looking for a segmentation of words into morphemes that minimizes the total length of the description of the dictionary of morphemes and the corpus of texts. Morphological analyzer Linguistica proposed by [Goldsmith 2001] was one of the first unsupervised models and was based on the MDL approach.

The Maximum Likelihood based method builds a probabilistic model of language morphology and tries to find a segmentation that maximizes the probability of the observed data (corpus of texts). Morfessor, one of the important models in the field of unsupervised morphology, was based on the maximum likelihood approach [Creutz, Lagus 2005].

K. Narasimhan, T. Kulkarni, R. Barzilay [Narasimhan et al. 2015] introduced an unsupervised method for uncovering morphological chains using a log-linear model that integrates both semantic and orthographic features. This approach builds a chain of possible word forms from a base word, outperforming the baseline Morfessor system in English, Turkish, and Arabic.

While there might be promising results in many scenarios, unsupervised models often lag behind supervised approaches in terms of accuracy and performance. This difference stems from the ability of supervised models to learn directly from explicit morphological annotations, leading to more precise analyses. Besides, handling rare and unseen words presents unique challenges. Unsupervised models, trained on the statistical regularities of the corpus, may struggle to accurately analyze words that occur infrequently or not at all in the training data. This can lead to errors or inaccurate morphological tag assignments, impacting overall performance.

The above-mentioned drawbacks bring us to the realm of supervised machine learning for morphological analysis. Supervised learning models operate under the assumption of a well-labeled dataset, where each word is accompanied by its corresponding morphological analysis, including lemma (a dictionary form), part of speech (a POS tag), and other relevant morphological features like gender, case, number, tense. etc. This approach allows models to learn explicit mappings between word forms and their morphological features. However, the reliance on labeled data presents a challenge, especially for low-resource languages where such resources may be scarce or entirely unavailable. Regardless of the language, the dataset has to be well documented as errors can affect the overall result. Creating these annotated datasets requires considerable linguistic expertise and is laborious.

Before training the model, the feature engineering has to be performed. Selecting and creating relevant features tailored to the specific language is crucial for maximizing

model performance. Once the features are defined, the next step involves choosing an appropriate classifier.

M. A. Kumar, Dhanalakshmi, K. P. Soman, S. Rajendran [Kumar et al. 2010] created a morphological analyzer for Tamil language based on sequence labeling using Support Vector Machine (SVM). The main idea of the sequence labeling approach is that the model accepts a sequence of characters as input and generates a sequence of characters as output. In their morphological analyzer, the input is a word denoted as 'W', and output is root and inflections denoted by 'R' and 'I' respectively. In this research, the dataset was created by classifying verb and noun paradigms based on tense and case markers respectively, grouping words sharing inflectional patterns. Preprocessing involved segmentation, breaking down words into graphemes and further into consonant-vowel (C-V) pairs, marked with '–C' and '–V' respectively. Segmented input and output word forms were aligned and mapped, creating training data pairs illustrating the mapping between word forms and their constituent morphemes (root and affixes). Authors developed separate engines for nouns and verbs. In this approach, they trained two models: one model is used for finding the morpheme boundaries, another model is used for assigning grammatical features to each morpheme.

T. Mueller, H. Schmid, H. Schütze [Mueller et al. 2013] address the challenges of training higher-order Conditional Random Fields (CRFs) for morphological tagging, particularly with large tag sets. Their work introduces a pruned CRF (PCRF) model, using a coarse-to-fine decoding strategy and early updating to achieve both speed and accuracy. This approach improves upon first-order CRFs across six languages, demonstrating the effectiveness of the PCRF for handling the complexities in combined Part-of-Speech (POS) and morphological (POS+MORPH) tagging.

A. Jayaweera, N. Dias [Jayaweera, Dias 2014] created a Part-Of-Speech (POS) tagger for Sinhala language using Hidden Markov Models. Sinhala is a morphologically rich and agglutinative language, which makes it challenging to automatically assign a tag to each word. The proposed POS tagger operates in two main steps: knowledge extraction from annotated corpus and tagging new text. During the first step, it processes a pre-tagged Sinhala corpus and calculates probabilities. For the second step, the Viterbi Matrix Analyzer constructs a state graph (Viterbi trellis) representing all possible tag sequences for the input words. It calculates and assigns state transition probabilities for each transition within this matrix. Then the tag sequence analyzer performs a backtrace through the Viterbi matrix to identify and output the most probable sequence of POS tags for the input word sequence. This method was applied to many languages like Arabic [Alajmi et al. 2011], Myanmar [Cing, Soe 2023].

*2.3. Deep learning based approaches*

While supervised and unsupervised machine learning methods have been applied to morphological analysis, they present significant challenges. Manual feature engineering is often laborious and requires linguistic expertise, and the reliance on annotated data for supervised approaches can be a major obstacle, particularly for low-resource languages.

Recent advancements in deep learning have led to the development of powerful neural network based morphological analyzers. As in the other NLP applications, so in the field of computational morphology deep learning models showed state of

the art results, outperforming earlier approaches [Liu 2021]. Moreover, one of the advantages of deep learning based methods is that they don't require predefined linguistic rules or high-level feature engineering. Architectures such as feedforward, Recurrent Neural Networks (RNNs), Convolutional neural networks (CNNs), Long Short-Term Memory (LSTMs), encoder-decoder structure and the transformer were applied for this task. In some approaches, neural networks systems were combined with statistical models such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) [Tamburini 2016].

Depending on how many output characters are expected relative to the number of input characters, morphological tagging can be conducted as the many-to-one learning scenario or as the one-to-one learning scenario [Bjerva 2017]. In the many-to-one learning scenario, the model gets multiple input symbols and then generates an output with only one symbol or one chunk of symbols like a POS tag or Morpho-Syntactic Description (MSD). In the one-to-one scenario, the model gets the input text and generates a prediction for each corresponding symbol in the input. Morphological tagging is often conducted on words in sentences as the use of information from context which is useful for good performance (Figure 1).

Morphological analysis can also be referred to as joint learning of lemmatization and morphological tagging. Lemmatization and morphological tagging are interdependent and can provide information to each other when there are different ways for lemmatization or tagging. [Müller et al. 2015] in their work have shown that modeling lemmatization and tagging jointly benefits both tasks. Following this method, [Heigold et al. 2017], combining a POS tag and MSDs together, explores neural character-based morphological tagging for languages with rich morphology and large tag sets. Authors compare two architectures for computing character-based word vectors using recurrent (RNN) and convolutional (CNN) networks. RNN architecture seems to perform as well or better than a CNN based architecture.

Another popular work based on joint learning is LemmaTag developed by [Kondratyuk et al. 2018]. Their neural network architecture that jointly generates part-of-speech tags and lemmas for sentences by using bidirectional RNNs with character-level and word-level embeddings. Their model consists of three parts:

1. the shared encoder, which creates embeddings for each word based on its character sequence and the sentence context;
2. the tagger encoder, which applies a fully-connected layer to predict the tags;
3. The lemmatizer encoder, which applies an RNN sequence decoder to the outputs of the shared encoder and the tagger.

They demonstrated that modeling lemmatization and POS tagging jointly by sharing the word and character embeddings and RNN encoder weights is effective for both tasks in morphologically complex languages. The results are higher in accuracy and the training requires less time. If the subcategories exist for the language (especially, for morphologically rich languages), LemmaTag also predicts each tag subcategory and inputs this information to the lemmatizer, which can further improve its accuracy. Besides, their model is featureless, requiring no text preprocessing and post-processing of morphological analysis. Figure 2 presents the architecture of their model.

R. Cotterell, G. Heigold [Cotterell, Heigold 2017] proposed a method that improves performance for low-resource languages through cross-lingual training on a related

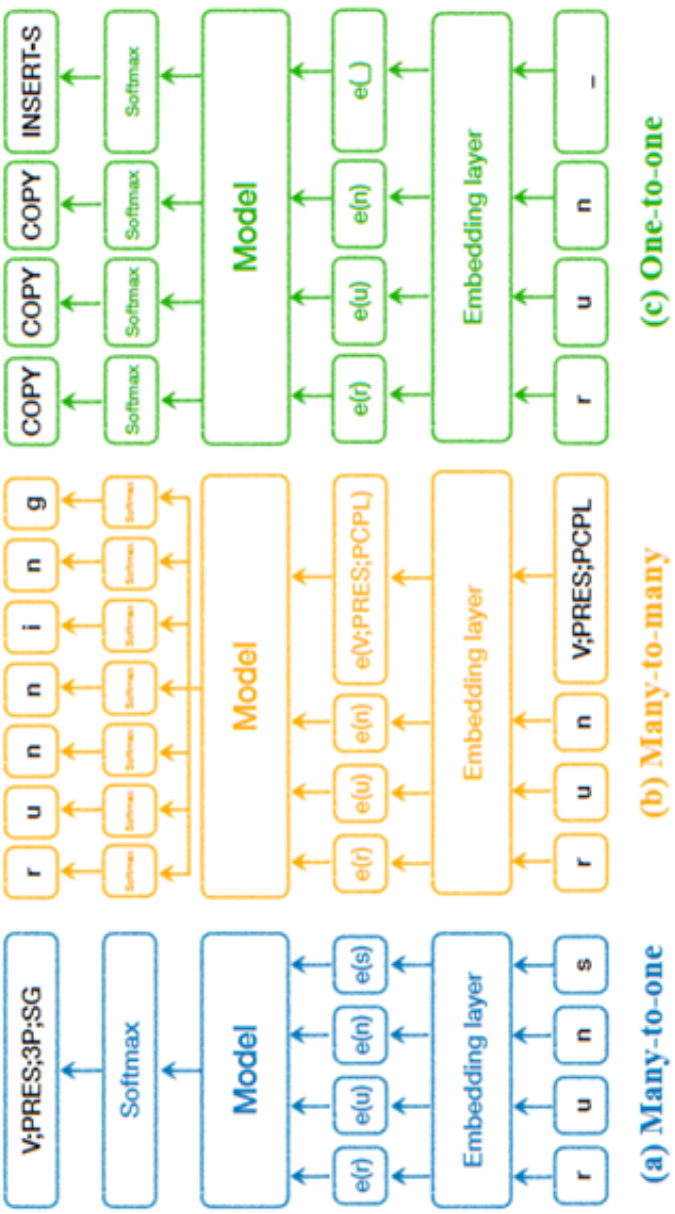


*Fig. 1.* Different neural network learning scenarios [Liu 2021]

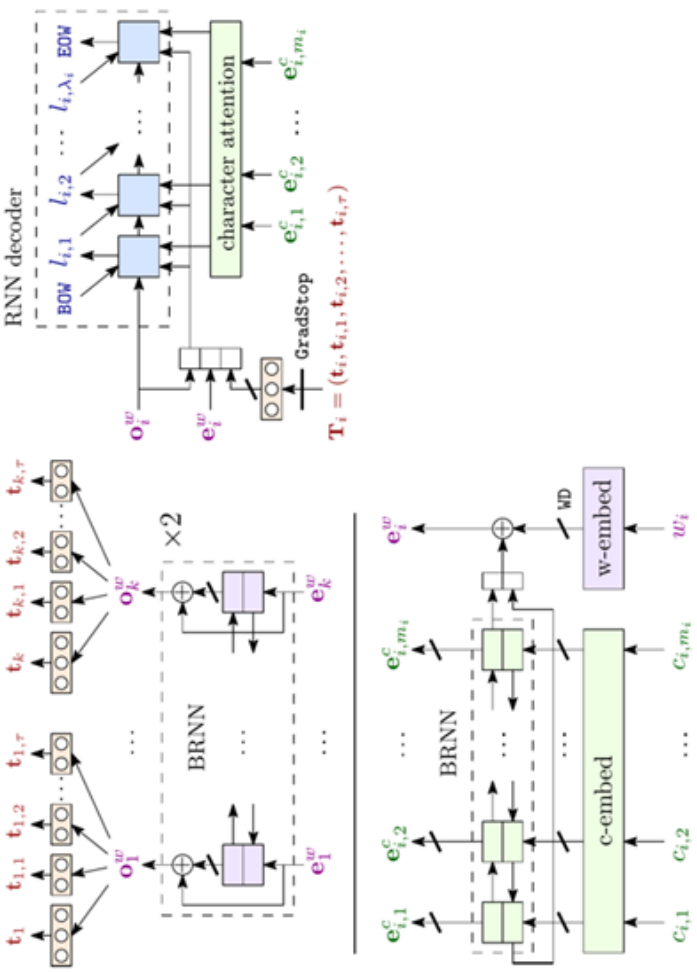


*Fig. 2.* Overview of the LemmaTag's architecture and design

high-resource language. However, this approach relies on the fact that target and source languages must be close and MSDs tag sets must match. Malaviya et al. (2018) trained a factorial conditional random field (FCRF) with neural network potentials calculated by LSTM. Their model makes predictions for individual tags separately, but links each decision together by modeling variable dependencies between tags over time.

E. Akyürek, E. Dayanık, D. Yuret [Akyürek et al. 2019] introduced a morphological analyzer called Morse. The proposed model uses three distinct encoders to create embeddings of various inputs. First encoder is a word encoder which creates word embeddings based on its characters. Second encoder is a context encoder which creates the local context embedding for each word based on the word embeddings of all words. Third encoder is an output encoder that creates an output embedding using the morphological features of the last two words. Then these embeddings are fed into the decoder to predict the lemmata and the morphological features.

G. Abudouwaili, K. Abiderexiti, N. Yi, A. Wumaier [Abudouwaili et al. 2023] created a joint morphological tagger for low-resource agglutinative languages to solve the challenges related to rule-based and statistical approaches such as error propagation, the reliance on linguistic expertise, missing the context features. First of all, authors represent the input by word embeddings, morphological embeddings or character-level embeddings, contextual embedding. The local context and word embeddings were generated by a pretrained language model. Authors used the pretrained cross-lingual language model XLM-RoBERTa for Kazakh, Tatar and Yak. This allows a model to better grasp the complex structure of agglutinative words and their role in the sentence. Second, the effect of errors in determining parts of speech on determining morphological features is reduced since the model simultaneously learns to predict a POS tag and MSDs. In their work, authors used a fusion mechanism, which allows two tasks (POS tagging and MSD tagging) to exchange information and influence each other in the process (Figure 3). The output of the fusion mechanism is fed into the CRF layer to predict the POS tag for each word. To predict MSD labels authors calculate label co-occurrence statistics and use a dynamic adjacency graph and GCN to find more deep label relationships.

D. Kondratyuk [Kondratyuk 2019] developed a morphological analyzer for the SIGMORPHON 2019 Shared Task on Morphological Analysis in Context and Cross-Lingual Transfer for Inflection, in task 2, Morphological Analysis in Context [McCarthy et al. 2019]. Authors utilize the pretrained multilingual BERT cased model to encode input sentences and to apply additional word-level and character-level LSTM layers. Then the lemma and the morphological features are jointly decoded. Lemmatization is treated as neural sequence tagging, and authors apply a feedforward layer to the final layer of the lemmatizer LSTM. Similarly for morphology tagging, they apply a feedforward layer to jointly predict the classes of unfactored and factored morphology tags. Figure 4 presents the architecture of their system.

Another transformer based work is [Wróbel, Nowak 2022]. For part-of-speech and morphological tagging, authors use XML-RoBERTa large, and for lemmatization a ByT5 small model was employed. The transformer returns the contextual embeddings of each token, and then the linear level with softmax activation returns normalized scores for each tag.

Another participant system for the SIGMORPHON 2019 Task 2 is Morpheus [Yildiz, Tantuğ 2019]. Their system is based on a neural sequential architecture

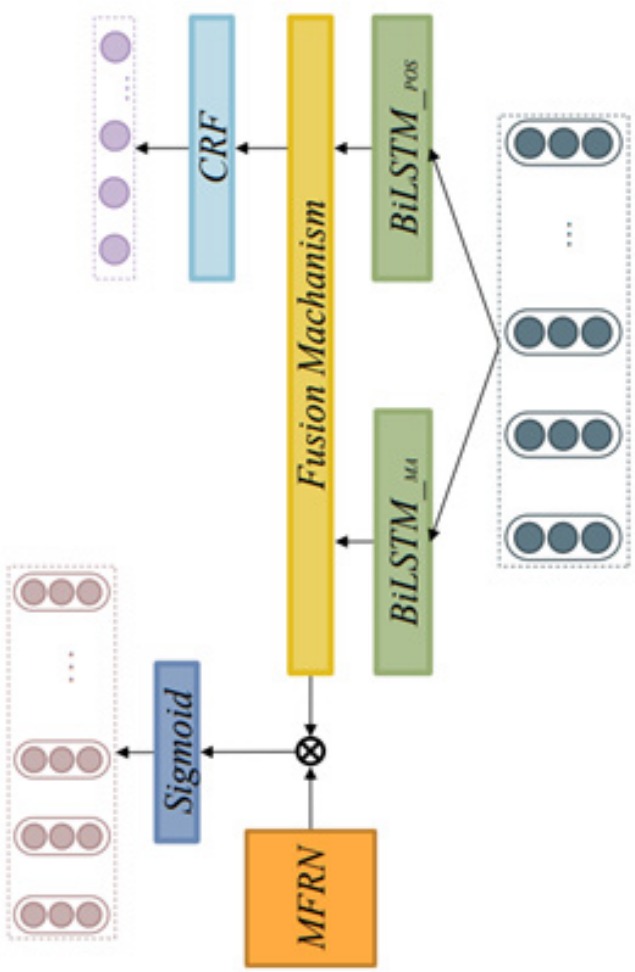


*Fig. 3.* The overall model architecture with fusion mechanism [Abudouwaili et al. 2023]

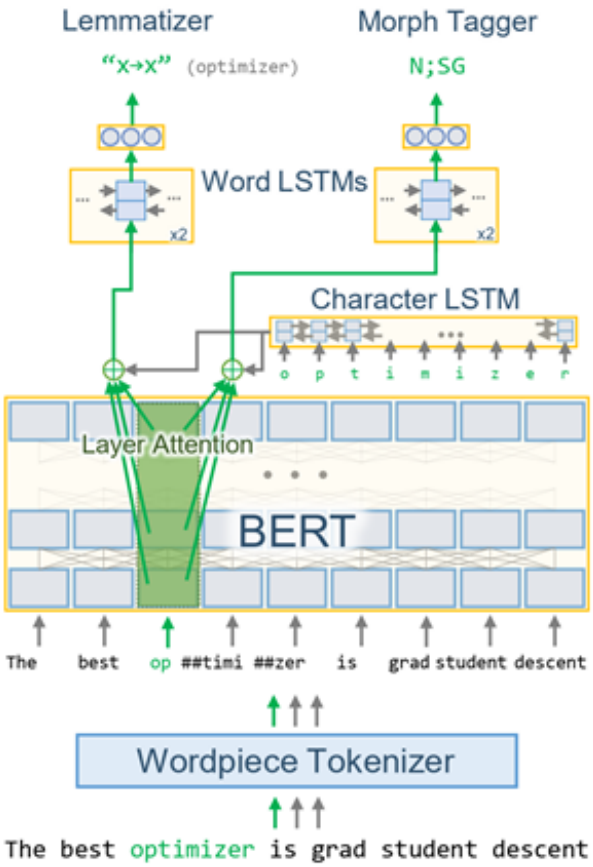


*Fig. 4.* An illustration of the model architecture based on BERT [Kondratyuk 2019]

where inputs are the sentences containing surface form words and the outputs are edit operations between surface words and their lemmata and morphological tags assigned to the words. To generate context-aware representations of each word, a bidirectional GRU is applied to characters of words and to the word representations. Two separate GRU decoders input the same context representation to generate transformations between the surface and root forms and to construct a morphological analysis. Figure 5 illustrates their proposed neural network architecture. The proposed model predicts minimum edit operations to obtain the lemma from a surface word, using a dynamic programming approach based on Levenshtein distance. The experiments carried out showed that predicting edit actions instead of characters in the lemma is higher in accuracy for both tasks. Morpheus does not require any language specific settings or pretrained embeddings so it is able to perform both tasks regardless of the language. Their system performs lemmatization and morphological tagging tasks comparable to state-of-the-art systems in almost 100 languages. According to the [SigMorphon 2019] Shared Task 2 results, Morpheus ranked 3rd in lemmatization and 9th in morphological tagging.

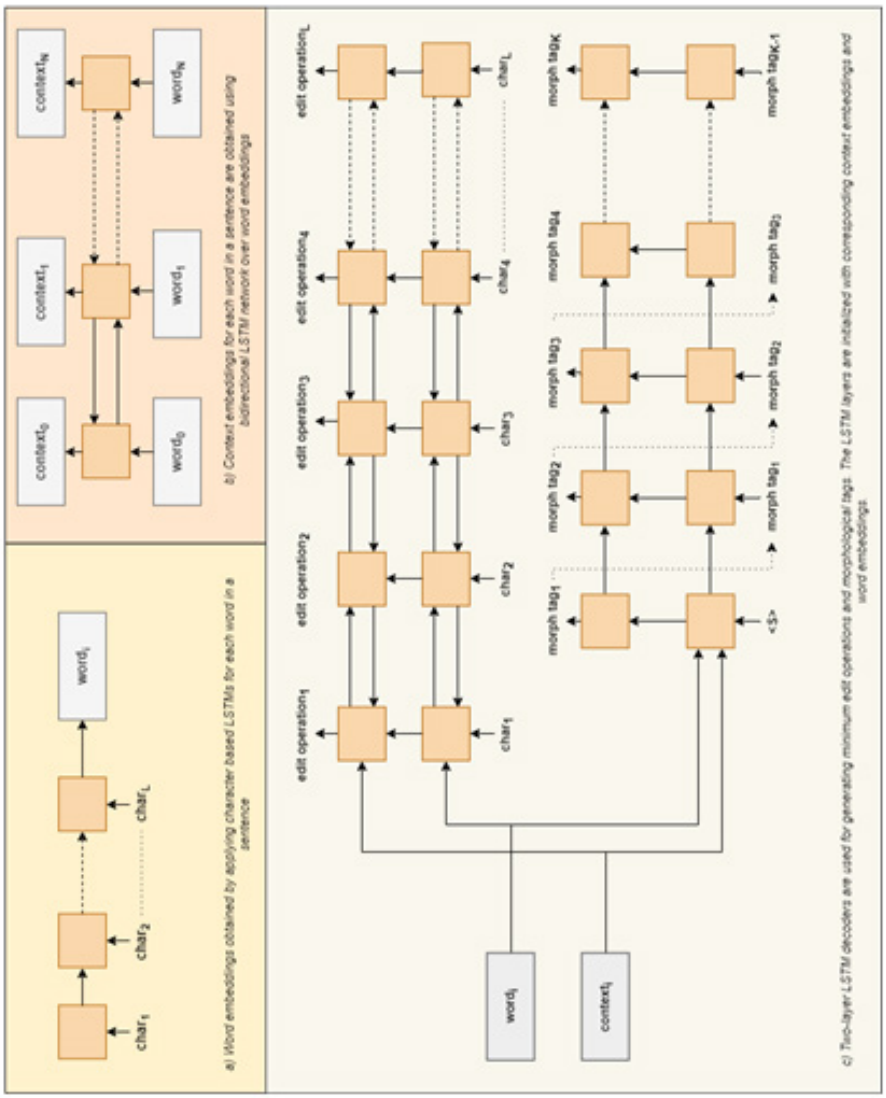


*Fig. 5*. The architecture of Morpheus [Yildiz, Tantuğ 2019]

**3. Data**

To create a dataset, a corpus comprising 146 Kalmyk folklore texts was used. These texts were obtained from the National Corpus of the Kalmyk Language[1] and subsequently converted into a format closely aligned with the standards of the Universal Dependencies[2] (UD) framework. In this format, individual sentences are delimited by blank lines, and each sentence begins with a comment line prefixed by '# text ='. This is followed by word lines, a sequence of lines, each representing the morphological analysis of a single token.

Word lines contain the following fields:

1. Word ID: Word index presenting the order of the corresponding word in the sentence.
2. Form: Word form or punctuation symbol.
3. Lemma: A dictionary form (lemma).
4. POS: A part-of-speech tag.
5. Morphological tags: List of morphological features.

Each token in the sentence is annotated with the word form, lemma, part of speech, and a set of morphological features. These elements are separated by tab characters. In cases where multiple morphological features are present, they are delimited by the pipe symbol (|). The "NoFeat" tag indicates that the token has no morphological tags (Figure 6).

Where possible, the original part-of-speech and morphological tags were mapped to the UD schema to improve compatibility. However, due to the specific characteristics of the Kalmyk language and the tagging annotations used in the source corpus, a substantial part of the annotation maintains its original tagset. For example, we cannot replace the 'N-N' tag with the regular 'N' tag. In Kalmyk, they differ and refer to other parts of speech. As a result, the final dataset is partially aligned with UD while preserving language-specific features. To train the Morpheus and our model, the MSD tags have been converted to the UniMorph schema, and the other six remaining columns were replaced with underscore (Figure 7).

The resulting dataset comprises 3,119 sentences and a total of 42,667 tokens. It includes 25 distinct part-of-speech tags and 135 unique morphological features. The dataset was divided into training, validation, and test subsets using an 8:1:1 split, respectively. All preprocessing and formatting were performed using custom Python scripts. The statistics of the dataset are shown in Table 1.

*Table 1.* Dataset statistics

| Dataset | Train | Valid | Test |
|---|---|---|---|
| Sentences | 2 495 | 311 | 313 |
| Tokens | 35 049 | 3 991 | 3 627 |

**4. Method**

The input embedding layer consists of two parts: word embedding and contextual embedding. Word embeddings are based on the characters that make up the word. This helps to take into account the morphology. Context embeddings are generated

[1] Национальный корпус калмыцкого языка [электронный ресурс] // URL: http://www.kalmcorpora.ru/ (дата обращения: 05.05.2025).

[2] CoNLL-U Format [электронный ресурс] // URL: https://universaldependencies.org/format.html (дата обращения: 05.05.2025).

```
# text = Чикән көндәхлә,тернь бас дурана.
1  Чикән  чикн  N  CASE=ACC1|PART=REFL
2  көндәхлә  көндәх  V  CONV=COND
3  ,  ,  PUN  NoFeat
4  тернь  тер  PRON  CASE=NOM|PART=POS3
5  бас  бас  CONJ  NoFeat
6  дурана  дурах  V  TENSE=PRES
7  .  .  PUN  NoFeat
```

*Fig. 6.* An example of annotated data

```
# text = Хальмг улс атхм чимг цуглулад,үүдн деерәсн уйна.
1   Хальмг  хальмг  _  _  ADJ  _  _  _  _
2   улс  улс  _  _  N;CASE=NOM  _  _  _  _
3   атхм  атхм  _  _  ADJ  _  _  _  _
4   чимг  чимг  _  _  N;CASE=ACC2  _  _  _  _
5   цуглулад  цуглх  _  _  V;VOICE=CAUS2;CONV=ANT  _  _  _  _
6   ,  ,  _  _  PUN  _  _  _  _
7   үүдн  үүдн  _  _  N;CASE=NOM  _  _  _  _
8   деерәсн  деер  _  _  POST;CASE=ABL;PART=REFL  _  _  _  _
9   уйна  уйх  _  _  V;TENSE=PRES  _  _  _  _
10  .  .  _  _  PUN  _  _  _  _
```

*Fig. 7.* An example of annotation where the MSD tags are converted to the UniMorph schema

by using pretrained cross lingual model XLM-RoBERTa[1] to take into account the word's meaning in the context. The decoder inputs a context representation of a word and applies GRU to generate the characters in the root form and the tags. The model utilizes one encoder for lemmas and MSDs tags, and two separate decoders to generate sequences of characters in lemmas and tags. Another decoder is used to input a context representation of a word and apply GRU to predict the sequence of transformation operations (for example, "insert a character", "delete a character", "replace a character") that must be applied to the source word (surface word) in order to obtain its lemma.

**5. Experimental results and analysis**

To compare and analyze the overall performance of our model, we found and trained several models that were also designed to jointly predict lemmas and morphological tags. It was important for us to choose models that would not depend on the language and the type of annotation. The selected models do not have to require additional morphological segmentation tools or pretrained embeddings.

To evaluate the models' performance, we selected the following metrics:

1. Lemma Accuracy: measures the percentage of words in the sentence for which the predicted lemma exactly matches the true lemma.

2. Lemma Levenshtein Distance: computes the Levenshtein distance between predicted lemmas and true lemmas. A lower value indicates better performance.

3. Morph Accuracy: evaluates the percentage of words where all morphological tags (POS + MSDs) are predicted correctly.

[1] XLM-RoBERTa (base-sized model) [электронный ресурс] // URL: https://huggingface.co/FacebookAI/xlm-roberta-base (дата обращения: 05.05.2025).

4. Morph F1: A harmonic mean of precision and recall for individual morphological tags.

The first model we trained was the neural network that is designed to jointly predict lemmas, part-of-speech (POS) tags, morphological tags. It combines character-level and word-level embeddings with a sequence-to-sequence (Seq2Seq) lemmatizer. Two bidirectional LSTMs were used to process character-level features and to generate word-level embeddings. Two linear layers were utilized to predict POS tags and MSDs. The Seq2Seq lemmatizer uses teacher forcing as a training mode.

Another proposed model is neural network, which combines character-level and word-level representations with Conditional Random Fields (CRF) for morphological tagging and a Seq2Seq lemmatizer. Both linear and CRF layers were used to predict part-of-speech tags and morphological features. This model has a flexible teacher forcing ratio which allows to balance exposure bias, and the CRF layers explicitly model dependencies between morphological tags.

Next model we chose to train was Morpheus [Yildiz, Tantuğ 2019]. Their system generates word embeddings and context-aware representations of each word in a sentence using bidirectional GRUs and then inputs two types of embeddings to decoder to generate lemmas and morphological features. Morpheus utilizes two decoder models, one model for lemmatization and the second one for morphological tagging. The outputs of this model are the minimum edit operations between surface words and their lemma and morphological tags of each word.

Table 2 shows the experimental results of several models and our model for the Kalmyk dataset. The proposed model demonstrates excellent results in the lemmatization task, outperforming all other models both in terms of accuracy (the highest) and the quality of lemma predictions (the lowest Levenshtein distance). Our model also shows best results in predicting morphological tags. Morpheus also has a good performance on lemmatization task, but is slightly inferior to our model in terms of accuracy and Levenshtein distance.

*Table 2.* Experimental results of morphological analyzers

| | Lemma Accuracy | Lemma Levenshtein Distance | Morph Accuracy | Morph F1 |
|---|---|---|---|---|
| Char-BiLSTM | 90,6 | 0,154 | 89,62 | 90,1 |
| Char-BiLSTM-CRF | 81,42 | 0,904 | 90,76 | 94,065 |
| Morpheus | 96,282 | 0,0493 | 90,041 | 92,938 |
| Our model | 96,66 | 0,047 | 91,71 | 93,604 |

## 6. Conclusion

This paper proposes a joint morphological analyzer based on a neural network for Kalmyk language. First, the model uses word embeddings based on characters of a word and context embeddings generated by pretrained cross lingual model XLM-RoBERTa to capture the morphology of the word and the local context. Second, our model is based on a sequential architecture which inputs surface words and predicts minimum edit actions between surface words and lemmas. We conducted several experiments to show that the proposed model outperforms other models trained for Kalmyk. In

future research, we will continue to improve the model's input embedding layer, joint learning of part-of-speech tagging and morphological features and the accuracy of lemmatization. The code of the proposed model can be found here.

**References**


Abudouwaili G., Abiderexiti K., Yi N., Wumaier A. Joint Learning Model for Low-Resource Agglutinative Language Morphological Tagging Proceedings of the 20th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology. Toronto: Association for Computational Linguistics, 2023. Pp. 27–37. DOI: 10.18653/v1/2023.sigmorphon-1.4 (In Eng.)

Akyürek E., Dayanık E., Yuret D. Morphological Analysis Using a Sequence Decoder Transactions of the Association for Computational Linguistics. 2019. No. 7. Pp. 567–579. DOI: 10.1162/tacl_a_00286 (In Eng.) (In Eng.)

Alajmi A. F., Saad E. M., Awadalla M. H. Hidden Markov Model Based Arabic Morphological Analyzer *International Journal of Computer Engineering Research*. 2011. No. 2(2). Pp. 28–33. DOI: 10.5897/ijcer.9000007 (In Eng.)

Baxi J., Bhatt B. Recent Advancements in Computational Morphology: a Comprehensive Survey. Available at: https://arxiv.org/pdf/2406.05424. 2024. Pp. 1–39. DOI: 10.48550/arxiv.2406.05424 (accessed: 05 May 2025) (In Eng.).

Baxi J., Patel P., Bhatt B. Morphological Analyzer for Gujarati Using Paradigm Based Approach with Knowledge Based and Statistical Methods. Proceedings of the 12th International Conference on Natural Language Processing. Trivandrum: NLP Association of India, 2015. Pp. 178–182. (In Eng.)

Bjerva J. One Model to Rule them all: Multitask and Multilingual Modelling for Lexical Available at: https://files.core.ac.uk/download/pdf/148336297.pdf. 2017. 266 p. DOI: 10.48550/arxiv.1711.01100 (accessed: 05 May 2025). (In Eng.)

Cing D. L., Soe K. M. Improving Accuracy of Part-of-Speech (POS) Tagging Using Hidden Markov Model and Morphological Analysis for Myanmar Language. *International Journal of Electrical and Computer Engineering (IJECE)*. 2023. No. 10(2). Pp. 2023–2030. DOI: 10.11591/ijece.v10i2.pp2023-2030 (In Eng.)

Cotterell R., Heigold G. Cross-lingual Character-Level Neural Morphological Tagging Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. Pp. 748–759. DOI: 10.18653/v1/d17-1078 (In Eng.)

Creutz M., Lagus K. Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0 ACM Transactions on Speech and Language Processing (TSLP). 2005. Vol. 4, Is. 1. Article No. 3. Pp. 1–34. DOI: 10.1145/1187415.1187418 (In Eng.)

Gebreselassie T. A., Washington J. N., Gasser M., Yimam B. A Finite-State Morphological Analyzer for Wolaytta Information and Communication Technology for Development for Africa. Vol. 244. Bahir Dar, 2018. Pp. 14–23. DOI: 10.1007/978-3-319-95153-9_2 (In Eng.)

Goldsmith J. Unsupervised Learning of the Morphology of a Natural Language Computational Linguistics. 2001. No. 27(2). Pp. 153–198. DOI: 10.1162/089120101750300490 (In Eng.)

Harris Z. S. From Phoneme to Morpheme Linguistic Society of America. 1955. Vol. 31. No. 2. Pp. 190–222. DOI: 10.1007/978-94-017-6059-1_2 (In Eng.)

Heigold G., Neumann G., Van Genabith J. An Extensive Empirical Evaluation of Character-Based Morphological Tagging for 14 Languages Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Vol. 1. Valencia: Association for Computational Linguistics, 2017. Pp. 505–513. DOI: 10.18653/v1/e17-1048 (In Eng.)

Jayaweera A., Dias N. Hidden Markov Model Based Part of Speech Tagger for Sinhala Language *International Journal on Natural Language Computing*. 2014. No. 3(3). Pp. 9–23. DOI: 10.5121/ijnlc.2014.3302 (In Eng.)

Kote N., Biba M., Kanerva J., Rönnqvist S., Ginter F. Morphological Tagging and Lemmatization of Albanian: A Manually Annotated Corpus and Neural Models Available at: https://arxiv.org/pdf/1912.00991. 2019. (accessed: 05 May 2025) (In Eng.)

Kondratyuk D., Gavenčiak T., Straka M., Hajič J. LemmaTag: Jointly Tagging and Lemmatizing for Morphologically Rich Languages with BRNNs Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. Pp. 4921–4928. DOI: 10.18653/v1/d18-1532 (In Eng.)

Kondratyuk D. Cross-Lingual Lemmatization and Morphology Tagging with Two-Stage Multilingual BERT Fine-Tuning. Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology. Florence: Association for Computational Linguistics, 2019. Pp. 12–18. DOI: 10.18653/v1/w19-4203 (In Eng.)

Koskenniemi K. Finite State Morphology and Information Retrieval Natural Language Engineering. 1996. No. 2(4). Pp. 331–336. DOI: 10.1017/s1351324997001587 (In Eng.)

Kumar M. A., Dhanalakshmi, Soman K. P., Rajendran S. A Sequence Labeling Approach to Morphological Analyzer for Tamil Language *International Journal on Computer Science and Engineering*, 2. 2010. No. 2(6). Pp. 1944–1951 (In Eng.)

Lindén K., Silfverberg M., Pirinen T. HFST Tools for Morphology — an Efficient Open-Source Package for Construction of Morphological Analyzers State of the Art in Computational Morphology. Vol. 41. Zurih, 2009. Pp. 28–47. DOI: 10.1007/978-3-642-04131-0_3 (In Eng.)

Liu L. Computational Morphology with Neural Network Approaches Available at: https://arxiv.org/pdf/2105.09404. 2021. DOI: 10.48550/arxiv.2105.09404 (accessed: 05 May 2025) (In Eng.)

McCarthy A. D., Vylomova E., Wu S., Malaviya C., Wolf-Sonkin L., Nicolai G., Kirov C., Silfverberg M., Mielke S. J., Heinz J., Cotterell R., Hulden M. The SIGMORPHON 2019 Shared Task: Morphological Analysis in Context and Cross-Lingual Transfer for Inflection. Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology. Florence: Association for Computational Linguistics, 2019. Pp. 229–244. DOI: 10.18653/v1/w19-4226 (In Eng.)

Mueller T., Schmid H., Schütze H. Efficient Higher-Order CRFs for Morphological Tagging. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Seattle: Association for Computational Linguistics, 2013. Pp. 322–332. DOI: 10.18653/v1/d13-1032 (In Eng.)

Müller T., Cotterell R., Fraser A., Schütze H. Joint Lemmatization and Morphological Tagging with Lemming. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics, 2015. Pp. 2268–2274. DOI: 10.18653/v1/d15-1272 (In Eng.)

Narasimhan K., Kulkarni T., Barzilay R. Language Understanding for Text-based Games Using Deep Reinforcement Learning Available at: https://arxiv.org/pdf/1506.08941. 2015. (accessed: 05 May 2025) (In Eng.)

Special Interest Group on Computational Morphology and Phonology. Available at: https://sigmorphon.github.io/workshops/2019/ (accessed: 05 May 2025) (In Eng.)

Tamburini F. A BiLSTM-CRF PoS-tagger for Italian Tweets Using Morphological Information. Proceedings of the 5th International Workshop EVALITA 2016. Napoli, 2016. Pp. 2531–4548. DOI 10.4000/books.aaccademia.1899 (In Eng.)

Wróbel K., Nowak K. Transformer-based Part-of-Speech Tagging and Lemmatization for Latin. Proceedings of LT4HALA 2022-2st Workshop on Language Technologies for Historical and Ancient Languages. Marseille: European Language Resources Association, 2022. Pp. 193–197. (In Eng.)

Zueva A., Kuznetsova A., Tyers F. M. A Finite-State Morphological Analyser for Evenki Language Resources and Evaluation. Marseille: European Language Resources Association, 2020. Pp. 2581–2589. (In Eng.)

Yildiz E., Tantuğ A. C. Morpheus: A Neural Network for Jointly Learning Contextual Lemmatization and Morphological Tagging. Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology. Florence: Association for Computational Linguistics, 2019. Pp. 25–34.(In Eng.)