

Правительство Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования

Национальный исследовательский университет
«Высшая школа экономики»

Факультет гуманитарных наук
Образовательная программа
«Фундаментальная и компьютерная лингвистика»

Картина Элен Геннадьевич

**Правильный морфологический парсер для шугнанского языка:
существительные, глаголы и прилагательные**

Выпускная квалификационная работа студента 4 курса бакалавриата группы БКЛ211

Академический руководитель образовательной программы
канд. филологических наук, доц.
Ю.А. Ландер

«_____» _____ 2025 г.

Научный руководитель
канд. филологических наук, доц.
Г.А. Мороз

Научный консультант
Стажёр-исследователь
М.Г. Мельченко

Москва 2025

Abstract

In this work I present a rule-based morphological analysis tool based on Helsinki Finite-State Technology (HFST) for the Shughni language (ISO: sgh; glottocode: shug1248), a language of the Iranian branch of the Indo-European family, a member of ‘Pamiri’ areal language group. While one existing rule-based parser exists for Shughni (Melenchenko, 2021), it does not utilize finite-state transducer technology. This work proposes the first HFST-based morphological parser implementation for Shughni, offering the advantages of this well-established framework for morphological analysis. The parser is presented in two variations: a morphological parser that breaks each word-form into stem and morphemes and assigns morphological tags to each one of them; a morphological generator that outputs word-forms taking a stem and morphological tags as an input. **TODO: prev sentence is questionable** This is a continuation my previous work, where nouns, pronouns, prepositions and numerals were implemented (Osorgin, 2024). This project covers **TODO: what**

TODO: Review abstract after finishing the work

Contents

1	Introduction	1
1.1	Shughni	1
1.2	Morphology modeling	2
2	Existing methods	2
2.1	Machine learning methods	2
2.2	Rule-based methods	3
2.2.1	Finite-state transducers	3
2.2.2	FST formalisms	4
2.2.3	Helsinki finite-state technology	5
2.3	Existing morphology models for Shughni	5
3	Data	6
3.1	Grammar descriptions	6
3.2	Dictionaries	6
3.3	Text corpora	6
4	Methods	8
4.1	Finite-state transducers TODO: Move to introduction?	8
4.2	Rule declaration TODO: rethink heading	8
4.2.1	Nouns	8
4.2.2	Verbs	8
4.2.3	Adjectives	8
4.2.4	Pronouns	8
4.2.5	Numerals	8
4.2.6	Anything else(?)	8
4.3	Transliteration	8
4.4	Russian lemmas TODO: rethink heading	8
4.5	Testing	8
4.6	Metrics	8
5	Results	8
6	Conclusion	8
	References	9

1 Introduction

1.1 Shughni

The Shughni language (ISO: sgh; Glottolog: shug1248) is a language of the Iranian branch of the Indo-European family (Plungian, 2022, p. 12). As of June 1997, it was estimated to be spoken by approximately 100,000 people (Edelman & Yusufbekov, 1999, p. 225) in the territories of Tajikistan and Afghanistan. Both countries have a subregion where Shughni is the most widely spoken native language. The Shughni-speaking subregion of Tajikistan is called ‘Shughnon’ and it belongs to ‘Gorno-Badakhshan Autonomous’ province. In Afghanistan, the Shughni-speaking region is called ‘Shughnan’ and it lies within the territory of ‘Badakhshan’ province (Parker, 2023, p. 2). Shughni belongs to ‘Pamiri’ areal language group, which is spoken along the Panj river in Pamir Mountains area.



Figure 1: Mountainous Badakhshan Autonomous Province of Tajikistan and Badakhshan Province of Afghanistan, (Parker, 2023, Fig 1.1)

There are three alphabets for Shughni that were derived from Cyrillic, Arabic and Latin scripts. Geographically the usage of said scripts correspond to the dominant script of each country where Shughni is spoken. In Tajikistan both official languages (Tajik and Russian) use Cyrillic script, so does Shughni on territory of Tajikistan. In Afghanistan Arabic script is used in Shughni, matching official languages (Pashto and Dari).

Latin script was developed and used in Tajikistan in 1930s (Edelman & Yusufbekov, 1999, p. 226) (Edelman & Dodykhudoeva, 2009, p. 788), but according to Edelman and Yusufbekov (1999) was not widely adapted. Later around 1980s a Cyrillic script gained popularity in Tajikistan, having some poetic literature and school materials based on Tajik’s alphabet, which is Cyrillic (Edelman & Yusufbekov, 1999). Today, Latin script is mostly used by researchers in scientific works.

The morphological parser developed in this work is based on materials that focus on Shughni spoken in Tajikistan. All the base lexicon is Cyrillic and comes from dictionaries that cover Shughni in ‘Gorno-Badakhshan Autonomus Province’. Latin script is supported with the help of transliteration.

1.2 Morphology modeling

Today there are two general approaches to the task of morphology modeling. The deep learning (DL) approach and the rule-based approach.

The DL approach typically makes use of training transformer models like BERT (Devlin et al., 2019) on vast amounts of marked-up data. This task becomes challenging, considering that Shughni is a low-resource language, meaning it lacks digital textual data. Although, DL approach was not utilized in this work, some existing DL approaches for low-resource languages are covered in section 2.1.

With the rule-based approach, morphological model is being built by writing grammar rules using some formalism language and listing base lexicon. In this work, rule-based approach was utilized, as it does not depend on the amount of available marked-up data as the DL approach does. It requires lexicons and morphological grammar descriptions, which exist for Shughni and which are discussed in Section 3.

2 Existing methods

2.1 Machine learning methods

There are a variety of LLM (Large language model) architectures that were applied to the task of language modeling. One significant example is LSTM (Long short-term memory) model, that was introduced by Hochreiter and Schmidhuber (1997). LSTM is a variation of RNN (Recurrent neural network), and it was widely applied to language modeling, including morphology modeling. Another more recent significant example is the transformer architecture presented by Vaswani et al. (2017), off which two years later BERT model was based (Devlin et al., 2019).

One of the biggest downsides of ML methods is that its quality depends on training data quantity, which makes it challenging to apply to low-resource languages such as Shughni. How-

ever, with introduction of LLMs this problem was shown to be solvable, for example, as shown by developers of UDify model (Kondratyuk & Straka, 2019). In their work authors show, that a BERT model pretrained on a large corpus of 104 languages can be fine-tuned on very little amounts of other languages’ data and still show decent results. For an example, they report that for Belarusian, UDify model achieved $UFeats = 89.36\%$ (accuracy of tagging Universal Features) after training on only 261 sentences from ‘Belarusian HSE’ Universal Dependencies treebank (Kondratyuk & Straka, 2019, Table 7).

However, working with LLM models is a highly resource-demanding task. The authors of UDify state, that the fine-tuning of their model for a new language would require at least 16 Gigabytes of RAM and at least 12 Gigabytes of GPU video memory, and the training process would take at least 20 days depending on the GPU model. While a deep learning approach would be interesting to explore, such computational resources are not available for this project. The neural approach is not the main target of this work and is implemented.

2.2 Rule-based methods

2.2.1 Finite-state transducers

The Rule-based approach historically is usually applied with the help of Finite-state transducers (FST), which is a variation of Finite-state machine, a mathematical abstract computational model. Following the terminology of Turing machines (Turing, 1937), a FST has two tapes: the input tape and the output tape. At any point it can read a next symbol from the input tape and then write a symbol to the output tape. Once a symbol was read from the input tape, it can not be read again, as the input tape shifts one symbol forward.

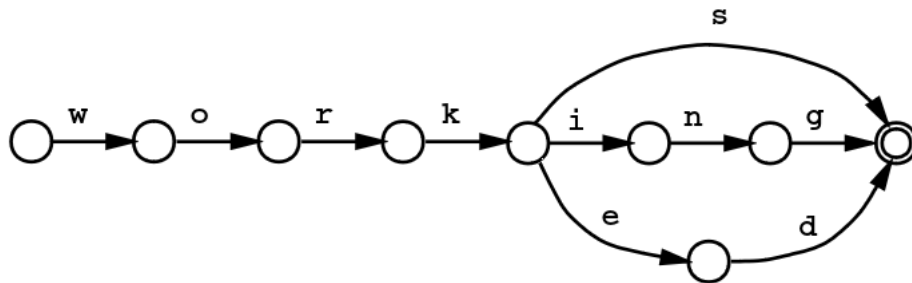


Figure 2: An example of FST with a single initial state (most left node) and a single final state (most right node) for a language where only three words exist: *works*, *working* and *worked*. The word *worker*, for example will not be recognized as a valid word by this FST, since there is no ‘d’ transition at state *worke*. The only way from *worke* state is via ‘d’ transition, which corresponds to the *worked* word. (Beesley & Karttunen, 2002)

The inner structure of FST can be illustrated as a directed graph with a set of all *states* (represented by graph’s nodes), a set of *transitions* (represented by graph’s edges), a set of *initial*

states (a subset of all the states, these are states where FST can start reading from the input tape) and a set of *final states* (a subset of all the states, these are the states where FST can stop reading from the input tape). A simplified FST is shown on Figure 2. The letters above the graph's edges denote *transition rules*, for an example *transition* 'w' means 'read *w* from the input tape THEN write *w* to the output tape'.

While working, FST will only make transitions that are possible from the current state. If there are no valid transitions then FST fails to process the input, and the input is considered to be impossible in the current language model. The measure of the amount of language's grammatical wordforms that successfully pass through the FST from an *initial state* to a *final state* will be called *Coverage* from now and on. The measure of the amount of language's ungrammatical wordforms that successfully pass through the FST from an *initial state* to a *final state* will be called *Overgeneration* from now and on. The ideal FST model of a language has a maximized 100% *Coverage* and a zero *Overgeneration*.

The model from the Figure 2 works effectively as a wordform paradigm dictionary, echoing back input wordforms that are grammatical and failing to output the whole ungrammatical wordforms. Now we can slightly adjust the transition rules in our example to make a morphological analysis tool that can be seen on the Figure 3. The notation of the *transition* 'w:w' dictates to read the left symbol from the input tape and write the right symbol to the output tape. If the spot on the right side is left empty, it means 'write nothing to the output tape'. An important note to remember is that FST can output only one symbol to the output while making a single transition. In this example '<inf>', '<pst>' and '<prs><2sg>' are 'multichar' symbols, meaning they are treated as three individual symbols by a FST, it will be covered in more detail in the Section 4.

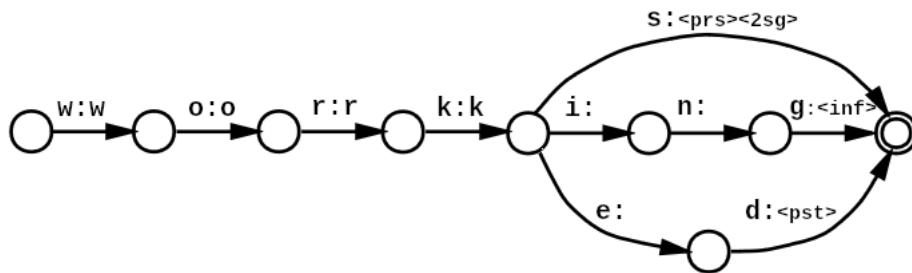


Figure 3: A modified version of Figure 2 which takes as input *works*, *working*, *worked* and outputs *work<prs><2sg>*, *work<inf>*, *work<pst>* respectively.

2.2.2 FST formalisms

By FST formalism I mean a human-readable formal language that can be compiled into a static FST, or from what a FST behavior can be emulated in runtime. A FST formalism usually

includes a way to list lexicons and/or list lexicon combination rules and/or list phonological rules.

One of the first major fundamental advances was when (Koskenniemi, 1983) created a model, which introduced a FST formalism named Two-level morphology (TWOL) for describing morphological and morphonological paradigms. Its novice was in the addition of the phonology level of rules, which made it much easier and intuitive to implement cases like ‘eye(-s)’/‘box(-es)’. This model was capable of word-form recognition and production, but it was not yet compilable into static FSTs, it was working at runtime and was known for being slow. Then Karttunen et al. (1987) at Xerox Research Center developed a Two-level rule Compiler (`twolc`), which compiled TWOL rules into static FSTs. Later a separate compiler for lexicon definitions was introduced named `lexc` (**Lexicon Compiler**) (Karttunen, 1993), it came with its own formalism language for describing lexicon and morphotactics. The standard approach to modeling a language at that point was using `lexc` to describe lexicon and morphology and `twolc` to describe morphonology, which stayed almost the same to this day.

One of the latest released tools was `lexd` lexicon compiler (Swanson & Howell, 2021). It is presented as a `lexc` alternative and is claimed to be much faster in the compilation time. It also introduced a tag system, which allows FST developers to specify how different lexicons combine with each other more precisely. **TODO: anything else?**

2.2.3 Helsinki finite-state technology

Helsinki finite-state technology (HFST) is a set of tools for creating and working with languages’ morphology models in form of transducers (Lindén et al., 2009). It includes implementations of both `hfst-lexc` and `hfst-twolc` compilers, as well as command line interface commands for mathematical and other miscellaneous operations with transducers like FST combination and format conversion. Also, it comes with a file format `.hfst` designed to store compiled FSTs.

HFST is widely applied when it comes to creating rule-based morphological models. Some of the latest examples of HFST-based morphological tools are: morphological parser for the Tamil language by Sarveswaran et al. (2021), a morphological transducer for Kyrgyz by Washington et al. (2012), a morphological parser for Andi by Buntyakova (2023) and a morphological parser for the Chamalal language by Budilova (2023).

2.3 Existing morphology models for Shughni

At this time only one morphological parser exists for Shughni. It was developed by Melnchenko (2021) and was later included in ‘Digital Resources for the Shughni Language’ project (Makarov et al., 2022). It is a rule-based parser implemented in Python which shows good coverage and accuracy results. The main difference from the parser presented in this work is that Melnchenko’s parser is not based on FST technology.

3 Data

3.1 Grammar descriptions

Several Shughni grammar descriptions were written throughout the years, starting from basic grammar description done by D. L. Ivanov (Salemman, 1895, pp. 274–281). An important mention is a work by Karamshoev (1963), which was the most detailed Shughni grammar description of its time. Latest significant works were ‘Shughni language’ (Edelman & Yusufbekov, 1999, pp. 225–242), ‘Comparative Grammar of Eastern Iranian Languages’ (Edelman, 2009) and ‘A grammar of the Shughhi language’ by Parker (2023), which is the biggest existing grammar, the most detailed and the most recent one.

For this work the main reference for compiling Shughni grammar rules was the work of Parker (2023). It was picked as it is the most recent, the most detailed and the biggest one. Other grammar description works were used too, but only as a secondary reference. The second most used grammar description was a work by Edelman and Yusufbekov (1999).

3.2 Dictionaries

There are two main dictionaries of the Shughni language: one by Zarubin (1960) and one by Karamshoev (1988–1999), both are written using Cyrillic script and include Russian translations. Some early dictionaries are ‘Brief grammar and dictionary of Shughni’ (Tumanovich, 1906), that is also using Cyrillic and translates to Russian, and ‘Shughni dictionary by D. L. Ivanov’ (Salemman, 1895), that translates to Russian but uses Arabic script alongside Cyrillic transcriptions for Shughhi word-forms.

An important lexical source for this work was the ‘Digital Resources for the Shughni Language’ project (Makarov et al., 2022). As a part of their work, authors compiled a digital dictionary for Shughni, where they digitalized both major Shughni dictionaries by Karamshoev (1988–1999) and Zarubin (1960). The digital dictionary is available at their website via a web-interface, but I was given access by the authors to a copy of the underlying database, which simplified the process of exporting lexicons for this project. All the lexicons for FST compilation were taken from their database.

3.3 Text corpora

I was given access to unpublished native texts that were gathered during HSE expeditions to Tajikistan in 2019-2024. It is not a large corpus of texts, its size can be seen on Table 1. The ‘Pear Story’ is a spoken text, it was written down from a retelling of the ‘Pear Story’ movie during an expedition.

Text name	Total tokens	Unique tokens
<i>'The Gospel of Luke'</i>	2978	1001
<i>'Pear Story'</i>	1117	438
Miscellaneous texts	164	106
All texts	4259	1393

Table 1: A list of native Shughni texts and their sizes gathered during HSE expeditions to Tajikistan in 2019-2024

The database provided by Makarov et al. (2022) also contained a lot of different useful data parsed from dictionaries including dictionary entries' usage examples. Such data is not as valuable as native texts, as sometimes it might not come from a native speaker but from a researcher. I would argue that for *Coverage* evaluation it might be quite useful.

From materials of HSE expeditions to Tajikistan I also acquired manually glossed texts in .eaf (ELAN) format. These texts were utilized for *Accuracy* evaluation, which will be discussed in Section 4.6.

A full list of text sources can be seen on Table 2.

Text name	Total tokens	Unique tokens	Native	Glossed
Dictionary examples	164 225	29 013	Uncertain	No
<i>'The Gospel of Luke'</i>	2 978	1 001	Yes	No
<i>'Pear Story'</i>	1 117	438	Yes	No
Miscellaneous texts	164	106	Yes	No
<i>'The Gospel of Luke'</i>	2 942	635	Yes	Yes
<i>'Pear Story'</i>	228	83	Yes	Yes
<i>'Mama'</i>	267	123	Yes	Yes

Table 2: A list of all available digital textual data

4 Methods

4.1 Finite-state transducers **TODO: Move to introduction?**

4.2 Rule declaration **TODO: rethink heading**

4.2.1 Nouns

4.2.2 Verbs

4.2.3 Adjectives

4.2.4 Pronouns

4.2.5 Numerals

4.2.6 Anything else(?)

4.3 Transliteration

4.4 Russian lemmas **TODO: rethink heading**

4.5 Testing

4.6 Metrics

5 Results

6 Conclusion

References

- Beesley, e. R., & Karttunen, L. (2002). *Finite-state morphology: Xerox tools and techniques*. CSLI, Stanford.
- Budilova, Z. A. (2023). *Создание морфологического парсера для чамалинского языка в системе lexd и twol [Morphological parser of Chamalal in lexd and twol]*, NRU HSE. <https://www.hse.ru/edu/vkr/837214661>
- Buntyakova, V. A. (2023). *Создание морфологического парсера андийского языка в системе lexd и twol [Morphological parser of Andi in lexd and twol]*, NRU HSE. <https://www.hse.ru/edu/vkr/837214826>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
- Edelman, D. I. (2009). Сравнительная грамматика восточноиранских языков [Comparative Grammar of Eastern Iranian Languages].
- Edelman, D. I., & Yusufbekov, S. (1999). Шугнанский язык [Shughni language]. In *Языки мира: Иранские языки. III: Восточноиранские языки [Languages of the world: Iranian languages. III. Eastern Iranian languages]*. <https://iling-ran.ru/web/ru/publications/langworld/volumes/7>
- Edelman, D. I., & Dodykhudoeva, L. R. (2009). Shughni. In G. Windfuhr (Ed.), *The iranian languages* (pp. 787–824). London & New York: Routledge.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Karamshoev, D. (1963). *Баджувский диалект шугнанского языка [Badzhuvskij dialect of the Shughni language]*. изд-во АН Тадж. ССР. <https://books.google.ru/books?id=8q1GXwAACAAJ>
- Karamshoev, D. (1988–1999). *Шугнанско-русский словарь [Shughni-Russian dictionary]*. Izd-vo Akademii nauk SSSR.
- Karttunen, L. (1993). Finite-state lexicon compiler. Technical Report ISTL-NLTT-1993-04-02, Xerox Palo Alto Research Center, Palo Alto, CA.
- Karttunen, L., Koskeniemi, K., & Kaplan, R. (1987). A compiler for two-level phonological rules. *tools for morphological analysis*.
- Kondratyuk, D., & Straka, M. (2019). 75 languages, 1 model: Parsing universal dependencies universally.
- Koskeniemi, K. (1983). Two-level Morphology: A General Computational Model for Word-Form Recognition and Production.
- Lindén, K., Silfverberg, M., & Pirinen, T. (2009). HFST Tools for Morphology – An Efficient Open-Source Package for Construction of Morphological Analyzers. In C. Mahlow & M. Piotrowski (Eds.), *State of the Art in Computational Morphology* (pp. 28–47, Vol. 41). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-04131-0_3
- Makarov, Y., Melenchenko, M., & Novokshanov, D. (2022). Digital Resources for the Shughni Language. *Proceedings of The Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-Resourced Languages in Eurasia within the 13th Language Resources and Evaluation Conference*, 61–64. <https://aclanthology.org/2022.euralli-1.9>

-
- Melenchenko, M. G. (2021). *Автоматический морфологический анализ шугнанского языка [Automatic full morphology analysis for Shughni]*, NRU HSE.
- Osorgin, I. G. (2024). *Создание морфологического парсера для шугнанского языка в системе lexd и twol [Creating a morphological parser for the Shughni language using lexd and twol systems]*, NRU HSE.
- Parker, C. (2023). *A grammar of the Shughni language* [Doctoral dissertation, Department of Linguistics, McGill University].
- Plungian, V. (2022). The study of shughni: The past and the future. *RSUH/RGGU Bulletin: "Literary Teory. Linguistics. Cultural Studies", Series. 2022;(5):11-22.*
- Salemann, K. (1895). Шугнанский словарь Д.Л. Иванова [Shughni dictionary by D.L. Ivanov]. In *Восточные заметки [Eastern notes]*.
- Sarveswaran, K., Dias, G., & Butt, M. (2021). ThamizhiMorph: A morphological parser for the Tamil language. *Machine Translation 35*, 37–70. <https://link.springer.com/article/10.1007/s10590-021-09261-5>
- Swanson, D., & Howell, N. (2021). Lexd: A Finite-State Lexicon Compiler for Non-Suffixational Morphologies.
- Tumanovich. (1906). *Краткая грамматика и словарь шугнанского наречия [Brief grammar and dictionary of Shughni]*.
- Turing, A. M. (1937). On computable numbers, with an application to the entscheidungsproblem. <https://doi.org/10.1112/plms/s2-42.1.230>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Washington, J., Ipasov, M., & Tyers, F. (2012). A finite-state morphological transducer for Kyrgyz. In N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the eighth international conference on language resources and evaluation (LREC'12)* (pp. 934–940). European Language Resources Association (ELRA). <https://aclanthology.org/L12-1642/>
- Zarubin, I. (1960). *Шугнанские тексты и словарь [Shughni texts and dictionary]*. Izd-vo Akademii nauk SSSR.