Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 5137–5142
Marseille, 20-25 June 2022

A Free/Open-Source Morphological Analyser and Generator for Sakha

Sardana Ivanova, Jonathan N. Washington, Francis M. Tyers
Helsingin yliopisto, Swarthmore College, Indiana University
Helsinki, 00014 Suomi, Swarthmore, PA 19081 USA, Bloomington, IN 47405 USA
sardana.ivanova@helsinki.fi, jonathan.washington@swarthmore.edu, ftyers@iu.edu

Abstract

We present, to our knowledge, the first ever published morphological analyser and generator for Sakha, a marginalised language of Siberia. The transducer, developed using HFST, has coverage of solidly above 90%, and high precision. In the development of the analyser, we have expanded linguistic knowledge about Sakha, and developed strategies for complex grammatical patterns. The transducer is already being used in downstream tasks, including computer assisted language learning applications for linguistic maintenance and computational linguistic shared tasks.

Keywords: morphology, Sakha, Turkic languages, FSTs, finite-state morphology, marginalised languages

1. Introduction

This paper describes the development of a morphological transducer for Sakha using free/open-source tools available as part of Helsinki Finite State Technology (HFST). Sakha is a Turkic language, with around 450 000 native speaker (Eberhard et al., 2022), primarily residing in the Sakha Republic, where the language enjoys official status. The Sakha Republic is located in Northeast Asia, and is part of the Russian Far East. Sakha speakers are subject to increasing economic (Streletskiy et al., 2019) and cultural (Lavrillier and Gabyshev, 2021) peril due to climate change, and the Sakha language faces social (Charter, 2022) and legal (Chevalier, 2017; Jankiewicz et al., 2020) marginalisation.

The transducer described in this paper provides morphological analysis and generation for Sakha, is entirely hand-crafted, and is publicly available under the GPL v3 Free/Open Source licence.[1] Morphological transducers can be used in a wide range of language technology applications and "downstream tasks"; e.g., they may be repurposed as spell checkers and used in rule-based machine translation pipelines. The Sakha transducer described here is currently used in Revita, a language learning application designed to support individual efforts at language maintenance (Katinskaia et al., 2018; Ivanova et al., 2019). Morphological transducers are an important technology for NLP, since they are linguistically informed and well understood, and require a single development cycle (Butt, 2020).

This paper is structured as follows. Section 2 gives an overview of the Sakha language and related work. Section 3 describes the methodology for implementing the transducer, including details of how the morphotactics and morphophonology were dealt with. Section 4 presents an evaluation of the transducer, showing that it has decent coverage and precision and recall. Finally, future work and conclusions are presented in sections 5 and 6, respectively.

2. Background and Methodology

2.1. Sakha

The Sakha language (also known as Yakut), is the language of nearly half a million speakers, mainly in the Republic of Sakha in the Far East of Russia (Siberia), shown in Figure 1. Sakha belongs to the Lena group of the Turkic language family, and its lexicon consists of Turkic words, borrowings from Mongolic and Tungusic languages, loanwords from Russian, and words of unclear (possibly Paleo-Asiatic) origin, (Kharitonov, 1987).

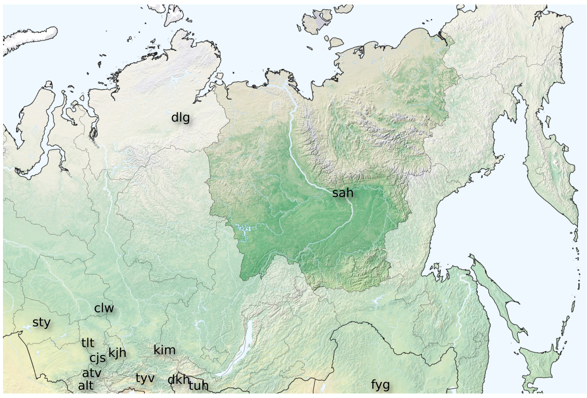

Figure 1: A map situating Sakha (`sah`) in Northeastern Asia and among its closest neighbouring Turkic languages (the remaining language codes, all ISO 639-3, aside from `dkh`, `fyg`, `tlt`, and `tuh`—used for Dukha, Fu-yü Gyrgys, Bachat Teleut, and Tuha, respectively). The Sakha Republic, where Sakha has official status, is highlighted.

Despite being rather divergent from other Turkic languages, Sakha shares a lot of properties with them: it can be described as agglutinating, meaning words may be inflected using a series of affixes; the word order is generally Subject-Object-Verb; and there are backness and rounding vowel harmony systems.

Various grammar sources were consulted in the development of the transducer, mostly Убрятова et al. (1982) and Sleptsov (2018).

[1] `https://github.com/apertium/apertium-sah`

2.2. Morphological transducers

The function of a morphological transducer is twofold: morphological generation takes surface forms (e.g., `атын`) as input and returns all possible lexical forms (e.g., `ат<n><px3sg><acc>/атын<adj>/атын<post>`, cf. /ɑt/ 'horse', /ɑtɯn/ 'different' (*adjective*), 'except' (*postposition*), and morphological analysis takes lexical forms (e.g., `ат<n><px3sg><acc>`) and returns one or more surface forms (e.g., `атын`). Morphological transducers are implemented as finite state transducers (FSTs), and in this case are compiled from hand-coded lexical, morphotactic, and morphophonological generalisations.

In this transducer, the lexicon and inflectional morphotactics are encoded in the `lexc` formalism, and the morphophonology in `twol`; the two files are compiled as FSTs using Helsinki Finite-State Technology (HFST) (Lindén et al., 2011), and these FSTs are intersected to produce the full transducer, per Koskenniemi (1983) and Beesley and Karttunen (2003). This follows the methodology used in previous Turkic FSTs (Washington et al., 2019).

3. Implementation

This section overviews the contents of the transducer and its design (§3.1), morphotactics (§3.2), and specific challenges (§3.3).

3.1. Contents and design

After initial implementation of the most straightforward morphology and lexical items, stems were added to the lexicon mostly by frequency of the occurrence of forms containing them in the Sakha Wikipedia corpus. Specifically, we went through an iterative process (documented in Washington et al. (2016)) of analysing the corpus using the transducer, identifying the stems of the most frequent unrecognised forms, adding those, recompiling, and running analysis again. This approach was adopted because it allowed us to increase coverage efficiently, starting with the most common forms.

Using this approach, the transducer reached a point where it included over 10 500 stems, consisting of around 5 400 nouns, over 2 100 proper nouns, over 1 300 adjectives, and over 1 000 verbs. The lexicon was recently expanded with words from Sleptsov (2018), increasing its size to over 37 000 stems, consisting of more than 12 400 nouns, 11 600 verbs, 5 700 adjectives, 3 900 adverbs, and 2 100 proper nouns. The remaining stems are divided between interjections, postpositions, numbers and numerals, conjunctions, determiners, pronouns, and other items (punctuation, abbreviations, etc.). Table 1 gives the number of lexical items for each of the major parts of speech.

The tagset consists of 105 separate tags, 15 covering the main parts of speech (noun, adjective, verb, adverb, postposition, etc.) and 90 covering lexical subcategory—e.g., transitivity, proper noun class, determiner type, etc.—and morphological function—e.g., case, number, person, possession, tense-aspect-mood, etc. The tags are based on the Apertium tagset[2] and are defined in the `lexc` source as multi-character symbols, between less than '<' and greater than '>' symbols, along with comments describing their usage.

Part of speech	Number of stems
Noun	12 423
Verb	11 686
Adjective	5 785
Adverb	3 971
Proper noun	2 135
Interjection	484
Postposition	143
Numbers	112
Conjunction	43
Determiner	33
Pronoun	16
Other	943
Total:	37 673

Table 1: Number of stems per part of speech

Regarding the phonology, rules in the `twol` formalism are applied in parallel, as contrasted with linearly ordered rewrite rules. While other formalisms allow for sequentially applied rules, we consider parallel rules to be preferable. The current `twol` file has 71 rules, the equivalents of which would be very difficult to order exactly right; we find it to be much more straightforward to simply ensure that no rules conflict with one another. We have encountered the belief that sequential rules are easier to work with, but we have not found that to be the case. Examples of the application of `twol` rules are discussed in detail in section 3.3.

3.2. Morphotactics

Morphotactics were implemented using the `lexc` formalism, through the use of continuation lexicons. For example, the `Root` lexicon points at the `Verbs` lexicon, which contains many verb lemma-to-stem mappings, each with the appropriate continuation lexicon (mostly either `V-IV` for intransitive verbs and `V-TV` for transitive verbs). Each of these continuation lexicons includes the relevant POS tags, as well as morphology-to-tag mappings which point to the appropriate following lexicon according to the morphotactics of the language. All the lexicons are concatenated as defined and provide variuos paths through the transducer. Every path must conclude with a terminal symbol, `#`. This approach works quite well for an agglutinative language with entirely suffixational morphology, like Sakha.

"Archiphonemes" are a type of control character used in the lexicon to represent morphology-level characters that may be realised in different ways orthographically depending on phonological alternations. For example, `{B}`, which begins many suffixes, may be realised as ‹б›, ‹м›, or ‹п› in different contexts, depending on the last character of a preceding morpheme. Other control characters may be used to trigger certain phonological practices without ever being realised orthographically themselves.

[2] `https://wiki.apertium.org/wiki/Tagset`

	harmonising vowel		
triggering vowel	high {I}	low {A}	diphthong {I}{A}
high	✔	✘	✔
low	✔	✔	✔
diphthong	✔	✘	✔

Table 2: The basic patterns of Sakha rounding harmony. ✔ indicates rounding, ✘ indicates no rounding. Harmonising high vowels and diphthongs always round after a round vowel trigger; harmonising low vowels only round after a round low vowel trigger.

3.3. Challenges

The implementation of the morphophonology of Sakha presented a number of challenges, described in this section: vowel harmony (§ 3.3.1), consonant assimilation (§ 3.3.2), and vowel epenthesis and other stem alternations (§ 3.3.3). Non-finite verb forms (§ 3.3.4) are also discussed.

3.3.1. Vowel harmony

The `twol` formalism for implementing two-level phonology is distinguished conceptually from other approaches to phonology: all rules are applied in parallel, are sensitive only to the input (morphological) form and the output (orthographic, in this case) form, and operate on only one character at a time. Two-level rules constrain a given input-output correspondence in one of several available ways given a particular context. Computationally, these rules are compiled into a finite-state transducer, and in our system (per Koskenniemi (1983) and Beesley and Karttunen (2003)), this FST is compose-intersected with the morphology FST, which maps analyses to morphotactic forms, as described in section 3.2.

Vowel harmony is exhibited in Sakha in two forms: backness harmony and rounding harmony. Vowels subject to harmony are encoded in the morphotactics with special symbols, conceptualised as "archiphonemes": `{A}` is a harmonising low vowel and `{I}` is a harmonising high vowel. In Sakha's system of rounding harmony, high and low vowels behave differently: low vowels only round after rounded low vowels, while high vowels round after any rounded vowel, high or low. Sakha's "falling" diphthongs, which orthographically consist of a high vowel component followed by a low vowel component, behave like high vowels, in that they round after any round vowel, and do not trigger the rounding of low vowels. This pattern is summarised in Table 2.

This general pattern of rounding harmony and the fact that long vowels (orthographically a sequence of two identical vowel characters) behave like their short vowel counterparts posed a challenge for `twol`. Since rules may only operate on a single character, each rule had to be sensitive to whether a harmonising vowel character is a component of a long vowel or a diphthong or not, and many of the alternations required multiple rules to implement.

An example of this is one of several rules conditioning the output of the `{A}` archiphoneme, needed so that it is realised in specific ways as the second character of a diphthong. This rule is shown in Table 2. Exceptions to this rule comprise environments where other rules affect `{A}` after `{I}`, and other rules that affect `{A}` might exclude the environment of this rule.

```
"{A} as part of {I}{A}"
%{A%}:Vy <=  %{I%}:Vx _ ;
         except
             :RealVow :RealVow %>: %{I%}:Vx _ ;
             _ %>: %{I%}: [ %{I%}: | %{A%}: ] ;
         where Vx in ( и ү ы у )
               Vy in ( э ө а о )
         matched ;
```

Figure 2: A `twol` rule affecting `{A}` when the second element of a diphthong. `{I}` harmonises as usual in these diphthongs, and the realisation of `{A}` is conditioned on that.

3.3.2. Consonant assimilation

Sakha also exhibits consonant assimilation processes that apply in both directions (anticipatory, in which an earlier segment is affected by a later segment, and persaverative, in which a later segment is affected by an earlier segment). Sometimes two assimilation processes involve the same consonants. For example, the verb form /tutn-bIt-A/ 'use–past.pfv–3' is realised as *тутуммута* [tutummuta], where the /n/ triggers nasalisation of the following /b/, and the /b/ triggers labialisation of the preceding /n/. This results in an /n-b/ sequence being realised as [mm].

In `twol`, because of the lack of rule ordering, this sort of mutual influence is not problematic, and may be implemented simply as two rules sensitive to the underlying form of adjacent consonants, as shown in Figure 3

```
"н→м assimilation"
Cx:м <=> _ (%{☭%}:) %>: %{B%}: ;
     where Cx in ( н %{н%} ) ;

"Assimilation after nasals"
Cx:Cy <=> :Nasal [ [ :0 - [ т: | Imaginary: | %{D%}: ] ] | %-: ]* _ ;
          %{м%}: [ [ :0 - [ Imaginary: | %{D%}: ] ] | %-: ]* _ ;
      where Cx in ( %{L%} %{T%} %{G%} %{B%} )
            Cy in (   н     н     ң     м   )
      matched ;
```

Figure 3: Rules in `twol` used to trigger separate anticipatory and perseverative assimilation processes which may interact with one another despite the lack of rule ordering. The underlying form of a symbol is the part to the right of the `:` characters. The `%` symbol is an escape character.

The morphological form of *тутуммута* is `тут{у}{н}>{B}{I}т>{A}`. The `{у}` is an epenthetic vowel, per §3.3.3, and `{н}` is a control sequence needed in place of an orthographic ‹н› to deal with passive forms of the verb.

It is important to note that this process of bidirectional assimilation is no more difficult to implement using rules applied in parallel than it is with sequential rules.

3.3.3. Vowel epenthesis and other stem alternations

There is a series of morphological stems in Sakha that end in two consonants, e.g. /usn/ 'swim', /køsn/ 'be visible', /oxn/ 'fall'. Since Sakha allows very few coda clusters (sequences of consonants at the end of a syllable), when these stems do not precede a vowel (which allows the second consonant to begin the next syllable), a vowel is inserted (or epenthesised) between the two final consonants, subjecting the first consonant to alternations due to being intervocalic. The stems above, for example, in isolation are *уһун* [uhun], *көһүн* [køhyn], and *оҕун* [oʁun], respectively, with /s/ and /x/ leniting to [h] and [ʁ]. Furthermore, when the second of the two consonants is able to syllabify as an onset, and no epenthetic vowel is needed, it is subject to desonorisation. In these three examples, the /n/ is realised as [t], as in the non-past forms *устар* [ustar], *көстөр* [køstør], and *охтор* [oxtor]. These forms are summarised in Table 3

A `twol` rule that maps an 'empty' character to another character is not guaranteed to work, since an empty transducer arc may not exist between two other segments, so the rule may have nothing to intersect with. To avoid this problem, the morphological form of these verbs, in `lexc`, was defined with a character (`{y}`) which, as specified by `twol` rules, is realised as empty when a vowel follows the consonant immediately after it, and which acts as a high harmonising vowel otherwise.

gloss	morphological representation	bare form	non-past
'swim'	`ус{y}н` /usn/	*уһун* [uhun]	*устар* [ustar]
'be visible'	`көс{y}н` /køsn/	*көһүн* [køhyn]	*көстөр* [køstør]
'fall'	`ох{y}н` /oxn/	*оҕун* [oʁun]	*охтор* [oxtor]

Table 3: Examples of stems with epenthetic vowels, and the other consonant alternations involved

Stem alternations like *уһун* 'swim.IMP' / *устар* 'swim-PRES' exhibit three single-character alternations in sequence, here: һ 'h'/с 's' due to intervocalic lenition, harmonised high vowel/∅ due to consonant cluster restrictions, and н 'n'/т 't' due to sonority restrictions. Each of these alternations required at least one `twol` mapping, variably sensitive to the other alternations and to other parts of the morphophonological context.

3.3.4. Non-finite verb forms

Sakha exhibits a high number of non-finite verb forms, many of which have finite uses as well. Previous grammars like Убрятова et al. (1982) categorise these forms as either participles or converbs, and do not present a detailed categorisation of their uses. As part of the construction of this transducer, we identified for each of these non-finite verb forms whether it had finite, verbal noun, verbal adjective, verbal adverb, or infinitive uses, and implemented each use separately. Many verb forms group into two categories: verbal nouns and verbal adjectives are often syncretic ("participles"), and verbal adverbs and infinitives (forms that occur with auxiliaries) are as well ("converbs"). However, we found that there is not a strict participle/converb binary as presented in previous sources. This work, documented in more detail in Washington and Tyers (2019; Washington et al. (2022), constitutes a novel understanding of Sakha grammar.

4. Evaluation

For evaluation, several corpora were prepared: a Sakha translation of the New Testament[3], a recent version of the Sakha Wikipedia [4], and a large newspaper corpus (Leontiev, 2015). The Wikipedia corpus was preprocessed to remove URLs, formulas, and lines containing Russian function words (which indicate non-Sakha content).

4.1. Coverage

The transducer[5] was evaluated in terms of naïve coverage, or the percentage of tokens receiving an analysis, whether correct or not. Mean ambiguity—the average number of analyses returned by the transducer per analysed token in a corpus—was also measured. These figures are reported for each corpus, along with the number of tokens, in Table 4.

Corpus	Tokens	Ambiguity	Coverage
New Testament	188K	2.50	94.53%
Wikipedia	2.4M	2.46	91.30%
Newspapers	16.0M	2.37	91.04%

Table 4: Naïve coverage and mean ambiguity of the analyser over several corpora

Before expanding the dictionary as described in (§3.1), the coverage on the Wikipedia corpus was somewhat higher than the Newspaper corpus; this was expected, since we added stems to the lexicon of the analyser based on a frequency list from the Wikipedia corpus. Adding more stems has brought the coverage of the two closer.

The ambiguity of all corpora is around or slightly lower than 2.5, meaning on average, every two analysed tokens in the corpus receive approximately five analyses total. Disambiguation (choosing between multiple analyses based on context) is a task for future work.

4.2. Accuracy

To understand the accuracy of the transducer, we calculated precision and recall. Precision is the percentage of analyses returned by the transducer for each form that are correct analyses of that form in some context. Recall is the percentage of the correct analyses for each form that are returned. These measures do not account for syntactic context.

To measure the transducer's precision and recall we selected 1000 valid words of Sakha randomly from the Wikipedia corpus, ran them through the analyser, and manually annotated this list to create a gold standard. Manual annotation consisted of adding analyses, and removing and

[3] `https://ibtrussia.org`

[4] `https://sah.wikipedia.org/`

[5] All evaluation for this paper was performed on revision `264da4c` from late March, 2022.

correcting returned analyses for each form. The gold standard was then compared to the original list of analyses returned by the analyser. Precision was 98.52%, recall was 75.42%; i.e., nearly every analysis returned by the transducer was deemed correct, but many correct analyses were not returned by the transducer (mostly due to low coverage). That is, many of the forms that caused low recall numbers are forms that were not analysed at all.

5. Future work

A number of minor issues in the implementation of some morphophonological alternations in the transducer were identified recently when the transducer was used as part of data generation for a shared task (Pimentel et al., 2021). Beginning to fix these has increased coverage slightly, and it is anticipated that further work on these and other minor morphophonology issues will also have some small impact on coverage. One example of this is the passive voice morpheme, the form of which is not predictable solely based on the orthographical form of the stem it attaches to. The form of the passive morpheme might need to be lexically specified or determined by control characters used in the lexicon.

Once good coverage has been achieved with a morphological analyser, the next logical step is to start work on morphological and syntactic disambiguation. As the mean ambiguity figures suggest, there is a lot of work that can be done on disambiguation.

6. Conclusion

We have presented, to our knowledge, the first ever published morphological analyser and generator for Sakha, a marginalised language of Siberia. The transducer has coverage of solidly above 90%, and high precision. In the development of the analyser, we have expanded linguistic knowledge about Sakha, and developed strategies for complex grammatical patterns. The transducer is already being used in downstream tasks, including computer assisted language learning applications for linguistic maintenance and computational linguistic shared tasks.

Acknowledgements

We would like to thank the Google Summer of Code 2018 for supporting the development of the Kazakh→Sakha MT system this transducer was designed for. We are very grateful to Igor Danilov and students of the Institute of Languages and Cultures of the Peoples of the North-East Russia for part-of-speech annotation of words from the dictionary. An earlier version of this work benefited from review and presentation at WiNLP 2021, and this paper benefited from review by three anonymous LREC reviewers.

7. Bibliographical References

Beesley, K. R. and Karttunen, L. (2003). Two-level rule compiler. `https://web.stanford.edu/~laurik/.book2software/twolc.pdf`.

Butt, M. (2020). Building resources: Language comparison and analysis. Invited talk at *The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Charter, D. (2022). Nivkh and Sakha language ideologies: Their causes and what they mean for language revitalization. Senior thesis, Swarthmore College.

Chevalier, J. F. (2017). School-based linguistic and cultural revitalization as a local practice: Sakha language education in the city of Yakutsk, Russian Federation. *Nationalities Papers*, 45(4):613–631.

David M. Eberhard, et al., editors. (2022). *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, online, twenty-fifth edition.

Ivanova, S., Katinskaia, A., and Yangarber, R. (2019). Tools for supporting language learning for Sakha. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa'19)*. Linköping University Electronic Press.

Jankiewicz, S., Knyaginina, N., and Prina, F. (2020). Linguistic rights and education in the republics of the Russian Federation: Towards unity through uniformity. *Review of Central and East European Law*, 45(1):59 – 91.

Katinskaia, A., Nouri, J., and Yangarber, R. (2018). Revita: a language-learning platform at the intersection of ITS and CALL. In *Proceedings of LREC: 11th International Conference on Language Resources and Evaluation*, Miyazaki, Japan.

Kharitonov, L. (1987). *Самоучитель якутского языка (Yakut language tutorial)*. Yakutsk Publishing.

Koskenniemi, K. (1983). *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. Number 11. University of Helsinki Department of General Linguistics, Helsinki.

Lavrillier, A. and Gabyshev, S. (2021). An indigenous science of the climate change impacts on landscape topography in Siberia. *Ambio*.

Leontiev, N. (2015). The newspaper corpus of the Yakut language. In *Proceedings of TurkLang 2015*, page 233.

Lindén, K., Axelson, E., Hardwick, S., Pirinen, T. A., and Silfverberg, M. (2011). Hfst—framework for compiling and applying morphologies. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 67–85. Springer.

Pimentel, T., Ryskina, M., Mielke, S. J., Wu, S., Chodroff, E., Leonard, B., Nicolai, G., Ghanggo Ate, Y., Khalifa, S., Habash, N., El-Khaissi, C., Goldman, O., Gasser, M., Lane, W., Coler, M., Oncevay, A., Montoya Samame, J. R., Silva Villegas, G. C., Ek, A., Bernardy, J.-P., Shcherbakov, A., Bayyr-ool, A., Sheifer, K., Ganieva, S., Plugaryov, M., Klyachko, E., Salehi, A., Krizhanovsky, A., Krizhanovsky, N., Vania, C., Ivanova, S., Salchak, A., Straughn, C., Liu, Z., Washington, J. N., Ataman, D., Kieraś, W., Woliński, M., Suhardijanto, T., Stoehr, N., Nuriah, Z., Ratan, S., Tyers, F. M., Ponti, E. M., Aiton, G., Hatcher, R. J., Prud'hommeaux, E., Kumar, R., Hulden, M., Barta, B., Lakatos, D., Szolnok, G., Ács, J., Raj, M., Yarowsky, D., Cotterell, R., Ambridge, B., and Vylomova, E. (2021). Sigmorphon 2021 shared task on morphological reinflection: Generalization across languages. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–259. Association for Computational Linguistics.

Sleptsov, P. (2018). *Саха тылын быһаарыылаах улахан тылдьыта: Большой толковый словарь якутского языка. [Large explanatory dictionary of the Yakut language: in 15 volumes].* Novosibirsk, Nauka.

Streletskiy, D. A., Suter, L. J., Shiklomanov, N. I., Porfiriev, B. N., and Eliseev, D. O. (2019). Assessment of climate change impacts on buildings, structures and infrastructure in the russian regions on permafrost. *Environmental Research Letters*, 14(2).

Washington, J. N. and Tyers, F. M. (2019). Delineating Turkic non-finite verb forms by syntactic function. In *Proceedings of the Workshop on Turkic and Languages in Contact with Turkic 4*, pages 132–146.

Washington, J. N., Bayyr-ool, A., Salchak, A., and Tyers, F. M. (2016). Development of a finite-state model for morphological processing of Tuvan. *Родной Язык*, 1(4):156–187.

Washington, J., Salimzianov, I., Tyers, F. M., Gökırmak, M., Ivanova, S., and Kuyrukçu, O. (2019). Free/open-source technologies for Turkic languages developed in the Apertium project. In *Proceedings of TurkLang 2019*.

Washington, J. N., Tyers, F. M., and Salimzianov, I. (2022). Non-finite verb forms in Turkic exhibit syncretism, not multifunctionality. *Special volume on multifunctionality and syncretism in non-finite forms*.

Е. И. Убрятова, et al., editors. (1982). *Грамматика современного якутского литературного языка: Фонетика и морфология [E. I. Ubryatova et al. Grammar of the modern Yakut literary language: Phonetics and morphology].* Москва: Наука.