A prototype finite-state morphological analyser for Chukchi

Vasilisa Andriyanets
School of Linguistics
Higher School of Economics
Moscow
bandrandr@yandex.ru

Francis M. Tyers
School of Linguistics
Higher School of Economics
Moscow
ftyers@hse.ru

Abstract

In this article we describe the application of finite-state transducers to the morphological and phonological systems of Chukchi, a polysynthetic language spoken in the north of the Russian Federation. The language exhibits progressive and regressive vowel harmony, productive incorporation and extensive circumfixing. To implement the analyser we use the well-known Helsinki Finite-State Toolkit (HFST). The resulting model covers the majority of the morphological and phonological processes. A brief evaluation carried out on publically-available corpora shows that the coverage of the transducer is between and 53% and 76%. An error evaluation of 100 tokens randomly selected from the corpus, which were not covered by the analyser shows that most of the morphological processes are covered and that the majority of errors are caused by a limited stem lexicon.

1 Introduction

This paper describes a new morphological analyser for Chukchi, an endangered language spoken on the Chukotka Peninsula in north of the Russian Federation (see Figure). The analyser is based on finite-state technology, which means that it can be used for both the analysis and the generation of forms — a finite-state morphological transducer maps between surface forms and lexical forms (lemmas and morphosyntactic tags).

An analyser of this sort has a wide variety of uses, including for automating the process of corpus annotation for linguistic research as well as for creating proofing tools (such as spellcheckers) and for lemmatising for electronic dictionary lookup for language learners — in a language with heavy prefixing and suffixing morphology, determining the stem is not a simple matter.

Our approach is based on the Helsinki Finite-State Toolkit (HFST, Linden et al. (2011)). We chose this toolkit over other toolkits such as `foma` Hulden (2009), as in addition to the `xfst` sequential rule formalism it also supports two-level phonological rules and weighted automata. We took an existing machine-readable dictionary and converted it to the `lexc` lexicon format, we then implemented the morphotactics (morpheme combinatorics) in `lexc` and used two-level (`twol`) rules for modelling phonological and some morphotactic constraints.

The remainder of the paper is laid out as follows: Section 2 gives a short introduction to Chukchi from a grammatical and sociolinguistic perspective; Section 3 describes other attempts at building a morphological analyser for Chukchi; Section 4 describes the methodology we used when building the transducer, including the tools used ; Section 5 describes in more detail what has been done and what problems arose when building the analyser; Section 6 describes a small evaluation and finally Section 7 contains our plans for future work.

2 Chukchi

Chukchi (in Chukchi: *лыгъоравэтльэн йилыйил* /ɬəɣʔorawetɬʔen jiɬəjiɬ/, ISO 639-3: `ckt`) is a highly-endangered minority language of the Russian Federation. It is spoken by around 5,000 people across the Chukotka Peninsula. Like the vast majority of other languages of the Russian Federation, intergenerational transmission is breaking down and there are few children learning the language. There are several relatively

Proceedings of Workshop on Polysynthetic Languages, pages 31–40
Santa Fe, New Mexico, USA, August 20-26, 2018.

Figure 1: Location of Chukchi-speaking area within the Russian Federation

similar spoken dialects and a written standard variety, which has been codified in a Soviet-era reference grammar, (Skorik, 1977) and differs from colloquial varities in a number of ways. The work described in this article is largely based on Dunn (1999). Dunn's grammar is written mostly for the Telqep dialect, which is spoken in the Tawajwaam village close to Anadyr, the administrative centre of Chukotka.

Chukchi is an ergative–absolutive language with rich morphology, both inflectional and derivational. In the nominal inflectional paradigm there are 13 cases and two numbers and in the verbal inflectional paradigm there are hundreds of forms. The language exhibits vowel harmony. Vowel harmony is a linguistic phenomenon where vowels are grouped and only vowels from the same group can occur in one word (cf. in most of the Turkic languages and some of the Uralic languages). That means, (almost) all vowels in a word can change if a certain affix is attached. The exact domain of vowel harmony can differ depending on the language.

In Chukchi vowels are split into two groups based on vowel height plus the schwa, which is neutral. Vowels in the first group, *э, о, а* /e$_2$, o, a/, are referred to as dominant, while vowels in the second group, *и, у, э* /i, u, e$_1$/ are referred to as recessive. Note that the difference between /e$_1$/ and /e$_2$/ is only a matter of harmony, they are pronounced the same. What is more, vowel harmony in Chukchi, unlike Turkic or Uralic languages, works both progressively and regressively and applies over the whole word. That is, morphological and phonological features of a suffix can cause vowel changes in the stem or vice versa — over any part of the phonological word. For example, a word *ӆыӆекэли* /ɬəɬekeɬi/ 'spectacled eider'[1] has recessive vowels, so if the dominant vowel harmony ablative case suffix *-гыпы* /ɣəpə/ is attached, the final form would be *ӆыӆякаӆэгыпы* /ɬəɬ**a**k**a**ɬeɣəpə/ 'from the spectacled eider' where harmonised vowels are indicated in bold. As can be seen in this example, the vowel *э* has both dominant and regressive readings and is both a dominant pair to /i/ and a regressive pair to /a/. This was one of the challenges for us while composing the transducer.

This is illustrated with some further examples. In example (1) the vowels in the stem *ӆыӆекэли* which are recessive are harmonised to the dominant variant after the addition of the ablative suffix *-гыпы* /ɣəpə/, which does not contain any dominant vowels (the two vowels are schwas), but in any case causes dominant harmony. Then in example (2), the essive case suffix vowel *-у* /u/ is harmonised to the dominant form *-о* /o/ after being attached to the stem *айван* /ajwan/ 'Eskimo' which has dominant vowels. Finally, in Example (3), the vowels in the stem *ээк-* /eek/ 'lamp' are harmonised to their dominant forms after the addition of the derivational suffix *-аӈӄача-* /aŋqatɕa/ 'side'.

(1)

Recessive	ӆ	ы	ӆ	е	к	э	ӆ	и	+	г	ы	п	ы	= ӆыӆекэли- + -гыпы
				↓		↓		↓						
Dominant	ӆ	ы	ӆ	я	к	а	ӆ	э	+	г	ы	п	ы	= ӆыӆякаӆэгыпы

(2)

Recessive	а	й	в	а	н	+	у	= айван- + -у
							↓	
Dominant	а	й	в	а	н	+	о	= айвано

[1]Lat. *somateria fischeri*, a large sea duck found in northeastern Siberia.

Agreement	Non-future	Example	Translation
1 Sg	т- **stem** -(гъэ)-к	*тэквэтгъэк* /tekwetɣʔek/	‘I leave’
1 Pl	мыт- **stem** -мык	*мытэквэтмык* /mətekwetmək/	‘We leave’
2 Sg	**stem** -(гъ)-и	*эквэтгъи* /ekwetɣʔi/	‘You leave’
2 Pl	**stem** -тык	*эквэттык* /ekwettək/	‘You leave’
3 Sg	**stem** -(гъ)-и	*эквэтгъи* /ekwetɣʔi/	‘She leaves’
3 Pl	**stem** -(гъэ)-т	*эквэтгъэт* /ekwetɣʔet/	‘They leave’

Table 1: Conjugation of intransitive verbs in the non-future/aorist tense, the verb has agreement markers for S, which denotes the syntactic role of a single-argument intransitive verb. Note the circumfix for the first person plural. Parentheses indicate optionality. Examples are based on the stem *-эквэт-* /ekwet/ ‘leave’.

		O					
		1 Sg	1 Pl	2 Sg	2 Pl	3 Sg	3 Pl
A	1 Sg	—	—	т- **stem** -гыт	т- **stem** -тык	т- **stem** -(гъэ)-н	т- **stem** -нэт
	1 Pl	—	—	мыт- **stem** -гыт	мыт- **stem** -тык	мыт- **stem** -(гъэ)-н	мыт- **stem** -нэт
	2 Sg	инэ- **stem** -(гъ)-и	**stem** -тку-гъ-и	—	—	**stem** -(гъэ)-н	**stem** -нэт
	2 Pl	инэ- **stem** -тык	**stem** -тку-тык	—	—	**stem** -ткы	
	3 Sg	инэ- **stem** -(гъ)-и	нэ- **stem** -мык	нэ- **stem** -гыт	нэ- **stem** -тык	**stem** -нин	**stem** -нинэт
	3 Pl	нэ- **stem** -гым				нэ- **stem** -(гъэ)-н	нэ- **stem** -нэт

Table 2: Conjugation of transitive verbs for agreement in the non-future/aorist tense. A is the semantic agent, or something that acts analogously and O is the semantic patient or anything else that acts analogously. For example, if we take the stem *-ӆыӷ-* /ɬʔu/ ‘see’ and we add the agreement morphemes *мыт-* /mət/ and *-гыт* /ɣət/ we get the form *мытӆыгыт* /mətɬʔuɣət/ ‘We see you’. Empty cells show impossible agreement combinations and parentheses indicate optionality. The table is adapted from (Dunn, 1999, p.177).

(3)

Recessive	э	э	к	+	а	ӈ	ӷ	а	ч	а	+	г	т	ы	= ээк- + -аӈӷача- + -гты
	↓	↓													
Dominant	а	а	к	+	а	ӈ	ӷ	а	ч	а	+	г	т	ы	= аакаӈӷачагты

Chukchi also has many morphophonological processes, which are expressed via mutations of letters in some contexts. This is further complicated by standard Chukchi orthography, which in some cases does not reflect the order of the sounds in a consistent manner.[2] For example, the glottal stop before a vowel at the beginning of a word or between two vowels is written as an apostophe *’* sign *after* the (second) vowel, while it is written as ъ or ь in other positions. Thus, the word *а’тчак* /ʔattɕak/ ‘to wait’ actually starts with the glottal stop when pronounced. This is further complicated by the fact that when prefixed, the glottal stop becomes a ь or ъ sign *before* the vowel, e.g. the neutral aspect aorist (non-future), first person singular agent, second person singular object form of *а’тчак* /ʔattɕak/ would be *мытъатчагыт* /mətʔattɕaɣət/ ‘I wait for you’. Figures 1 and 2 show an example table from Dunn (1999) for neutral aspect aorist for transitive and intransitive verbs respectively.

In terms of morphology, Chukchi inflectional morphology is both suffixing and prefixing, and in many cases circumfixing. Chukchi transitive verbs have A–O agreement and inflect for both subject and object,[3] but the combinations are not agglutinative and cannot be divided into “affixes for A” and “affixes for O”.

Chukchi derivational morphology is abundant and very productive, with both derivational prefixes, such as *рэ-* /re/ for desiderative and suffixes such as *-тку* /tku/ for iterative.

3 Related work

There was an attempt at making a morphological analyser for Chukchi verbs and nouns using Uniparser (Arkhangelsky, 2012). This used an approach to morphological analysis based on affix stripping with surface

[2]Our rationale behind building the transducer based on the official orthography is that we would like to have the possibility of producing a proofing tool, and the ease of treating Russian loanwords — which would be complicated by transcription.

[3]Here we follow the terminology of Dunn (1999) in labelling the subject of an intransitive verb as S, or ‘subject’ and the subject of a transitive verb as A or ‘agent’. Dunn (1999) defines S as “…the syntactic role of the single argument denoted by the syntactic valency of an intransitive verb” and states that “…A and O are distinguished from S in that they are with reference to the syntactic valency of a transitive verb”.

constraints using regular expressions (no underlying forms). While the system was able to analyse some part of the Chukchi noun paradigm, it was not able to deal with circumfixes, incorporation or long-distance morphological dependencies. We used the machine-readable lexicon to bootstrap our `lexc` lexicon.

4 Methdology

The transducer described in this paper is designed based on the Helsinki Finite State Toolkit (HFST, (Linden et al., 2011)), which is popular in the field of morphological analysis. It implements both the `lexc` formalism for defining lexicon and morphotactics, and the `twol` and `xfst` formalisms for modelling morphophonological rules. This toolkit has been chosen because it, or the related `foma` (Hulden, 2009), has been widely used for other agglutinative and polysynthetic languages, such as Navajo (Hulden and Bischoff, 2008), the Dene languages (Arppe et al., 2017), Quechua (Rios, 2016) and Arapaho (Kazeminejad et al., 2017), and is available under a free/open-source licence.

A finite-state transducer is a formal way to map surface forms and analyses (lexical forms) to one another. For example, *гэӈээккэтэ* /ɣeŋeekkete/ 'сом-daughter-сом' would receive the analysis ӈээккэт<n><com>.[4]

The transducer accepts the form as input and outputs the analysis, and vice versa. When used for modelling natural-language morphology, a finite-state transducer is a directed graph where the arcs encode relations between input symbols and output symbols. These symbols may be letters, linguistic tags or archiphonemes[5] Analysing or generating a form involves traversing the graph from left to right, while reading a symbol and outputting its corresponding symbol.

5 Implementation

The full transducer is composed of three transducers:

- `lexc` deals with morphology and lexicon;
- `twoc` helps to implement morphology that is not possible (or too difficult) to implement in lexc, e.g. certain circumfixes;
- `twol` deals with phonology and morphophonology.

In the following subsections we describe each of these components in turn.

5.1 Lexc

The dictionary we took roots from consisted of 6,321 lexemes, divided into 10 word classes. The resulting `lexc` file contains 15 continuation classes for nouns, one continuation class for verbs that led to either transitive or intransitive class, and several other classes to cover closed classes such as pronouns and conjunctions.

One of the drawbacks of the dictionary we used was that it did not include information on the transitivity of verbs — this causes a problem because transitive and intransitive verbs inflect very differently (see Tables 1 and 2). This meant that we had to guess the transitivity of a verb by its affixes. This kind of guessing is possible for finite forms (as the agreement suffixes differ between transitive and intransitive), but increases the size of the transducer and is impossible for non-finite forms (as they are not marked for agreement in the same way). As the transitivity of a verb form is fairly clear from the affixes, given a large enough corpus it should be possible to semi-automatically determine the transitivity of individual stems, however any guesses made this way would need to be proofread.

Not having the verbs marked for transitivity can cause two problems, the first being that we may analyse forms that do not exist (for example, an intransitive suffix on a transitive verb stem); the second being that these combinations may be homographous with word forms that do exist, resulting in an incorrect analysis and potential for *hiding* gaps in the lexicon.

To include circumfixes in the analysis, flag diacritics were used. This allowed the output tags for words with circumfixes to follow the Apertium (Forcada et al., 2011) tag style. Flag diacritics are a

[4]The tags used here mean 'noun' and 'comitative case'. See appendix 5 for a list of the tags used in this article.

[5]An archiphoneme is a symbol represents an underspecified phoneme which is determined by context; that is a phoneme which can have more than one surface realisation depending on context.

way of restricting certain combinations of discontinuous parts of words. They have the format @FLAGTYPE.FEATURE.VALUE@, where FLAGTYPE is the type of the flag, i.e. what it does: whether it sets the value of a feature (P flags), requires certain value of a feature (R flags) or demands that a feature has not been given value at all (D flags); FEATURE is the name of the feature one gives, and VALUE is the value of the feature that should or should not be matched with other flag diacritics with the same FEATURE or other FLAGTYPEs. For example, the verbal prefix used in the intransitive <s_sg1> form *мыт-* /mət/ sets the value of a feature VPR (verbal prefix) to myt: @P.VPR.myt@. Then, after the root, the suffix part of the <s_sg1> circumfix demands the value of the feature VPR to be set to myt: @R.VPR.myt@. Other suffixes demand either other prefixes or that the value of the feature is not set: @D.VPR@.

The tag style we used was that the lemma always preceded the tags, and the order of the tags was the following: first the category tags (like <v>); prefix derivations; suffix derivations; inflection. This posed a certain challenge for us and basically required treating every prefix in the same manner as a circumfix, as the morpheme appeared before the stem, but the tag appeared after stem. The system presented in Kazeminejad et al. (2017), where tags appear *in situ*, which could lead to a more efficient implementation, but less uniform tag strings. We made that choice based on considerations of tagset simplicity as opposed to implementation simplicity.

It was decided that all of the verbal inflection paradigm should be written with flag diacritics, as every inflected verb has some of the inflection affixes, and flag diacritics cut the number of branches in the resulting transducer down, making the transducer smaller. Some parts of prefixal and circumfixal verbal derivation, however, were moved to the twoc module, as the overflow of flag diacritics would make the source files unreadable.

5.2 Twoc

The twoc module included twolc-style rules for prefixes. The purpose of these rules was to be able to have the tag representing the feature to appear later in the string than where the morpheme appeared. For example, consider the form *ҷинэнимэтги* /qinenimetgi/ '(You) make me curdle!', which has the underlying form: ҷ{ы}>ин{Æ}>{R}{ы}>им{Æ}т>и (glossed: INTL.ASG2.OSG1.CAUS-curdle-ASG2.OSG1). In order to get the causative tag to appear after the stem we add markers in the tag string, enclosed in square brackets at the point at where the morpheme appears and at the point at which the tag appears,

Surface form:	ҷинэнимэтги
Morphotactic form:	ҷ{ы}>ин{Æ}>{R}{ы}>им{Æ}т>и
Lexical form:	[+caus]имэтык<v><tv>[+caus]<caus><neut><intn><a_sg2><o_sg1>

We then introduce a twol rule (see following) which forbids strings where only a single [+caus] marker appears (e.g. the prefix without the tag, or the tag without the prefix.

```
"Causative"
%[%+caus%]:0 <=>        _ :* %[%+caus%]:0 ;
        %[%+caus%]:0 :* _ ;
```

The difference between these rules and the flag diacritics mentioned previously is that they do not require processing at runtime.

5.3 Twol

The twol part of the transducer treated orthographic, phonologic and morphophonologic rules. In twol each rule is compiled into a transducer and all the rules are applied simultaneously.[6]

Two-level rules are equivalent in expressive power to sequential rules (Karttunen, 1993), however from the point of view of the linguist they have some differences particularly in how phonology is conceptualised.

[6]A reviewer suggests that xfst-style sequential rules are the dominant paradigm for implementing morphophonological rules. We would dispute this, while sequential rules are definitely popular, two large repositories of freely-available morphological transducers, Apertium (Forcada et al., 2011) and Giellatekno (Moshagen et al., 2014) are largely based on two-level rules, with Apertium having none out of 77 transducers based on sequential rules and Giellatekno having six out of 52.

```
"Vowel harmony"
! а й в а н >:0 у:о
Vx:Vy <=> [ Dominant | %{⁺ᵛᴴ%}: ] :* _ ;
                                    _ :* [ Dominant | %{⁺ᵛᴴ%}: ] ;
       except
                        .#. _ %>: :Vow ;            ! contexts for deletion
                       :Cns _ (:0) (:0) %>: :Vow ;  ! deletion; (:0) (:0) = special symbols
                            _ :*  %>: %{ӕ%}: ;      ! doesn't affect loan phonology
                            _ %{°%}:0 .#. ;         ! word-final abs reduction
                            _ %{·%}:0 .#. ;         ! word-final abs deletion
      [ :ч | :□ ] (:0) (:0) _ ;                     ! orthography
:Vow (:0) (:0) й: (:0) (:0) _ ;                     ! orthography
       where
           Vx in ( у ю и %{Æ%} )
           Vy in ( о ё э   а   ) matched ;
```

Figure 2: A `twol` rule to apply vowel harmony constraints. A vowel `Vx` is changed to `Vy` if there is any dominant vowel anywhere in the string, or if the special marker of dominant harmony `{⁺ᵛᴴ}` appears. The `except` contexts apply to loanwords, vowels which are deleted, and vowels which are processed by other rules because of orthographic rules, e.g. *и* /i/ should change to *е* /e/ and not *э* /e/ in certain contexts.

In two-level rules, the rules are viewed as constraints over a set of all possible surface forms generated by expanding the underlying forms using the alphabet, whereas in sequential rules, the rules are viewed as a sequence of operations for converting an underlying to a surface form. While it may not be relevant from an engineering point of view, we find conceiving of rules as constraints over all possible forms to be more cognitively plausible. Readers are encouraged to review Karttunen (1993) for a more thorough comparison of the techniques.

The most important set of rules was for vowel harmony. This involved dealing with both phonological considerations and orthographic considerations. First of all, all of the vowels in the word should agree for harmony class. Second, after the letter *ӆ* /ɬ/ in the Cyrillic orthography vowels are written with Russian iotated vowels, so that the sequences *ӆе* /ɬe/ and *нэ* /ne/ have the same vowel. Finally, Russian loanwords can contain otherwise impossible combinations of vowels and always behave as if they were dominant, i.e. they do not change. For example, in the word *округе*[7] there are two vowels (*о* /o/ and *у* /u/) which form a harmonic pair: *о* /o/ is dominant, while *у* /u/ is recessive, i.e. it usually changes in the presence of a dominant vowel or a dominant harmony marker. A special sign ({␣}) was introduced so as to not overwrite the vowels in such cases. The general-case vowel harmony rule is given in Figure 2. This rule turns vowels listed in the `where` section in the end as `Vx` to vowels listed there as `Vy` in certain contexts.

There is also a set of rules that govern syllable structure. Chukchi has a strict syllable structure: the maximal structure is CVC, usually CV. On one hand, if a consonant cluster of three or more consonants occurs, it is syllabified with a schwa, *ы* in the orthography. If multiple contexts co-occur, CV(C) syllable templates are assigned to sounds from right to left, and then schwas are inserted. An illustration of this process can be seen in Figure 3.

On the other hand, the onsets of syllables are always filled (unless the word starts with a vowel), so if two or more vowels occur next to each other, the first one is deleted.

Other phonological processes that were dealt with were vowel deletion, consonant deletion in certain contexts, affix allomorphy, vowel reduction, and archiphonemic allomorphy.

5.4 Results

The Figure 4 below shows the example output from the transducer.

So far, the analyser accounts for:

- the vast majority of morphophonology and orthography issues;
- nominal, pronominal, adjectival inflection and derivation; uninflected parts of speech;

[7] Rus. /о́круг/, an administrative division often translated as 'district'.

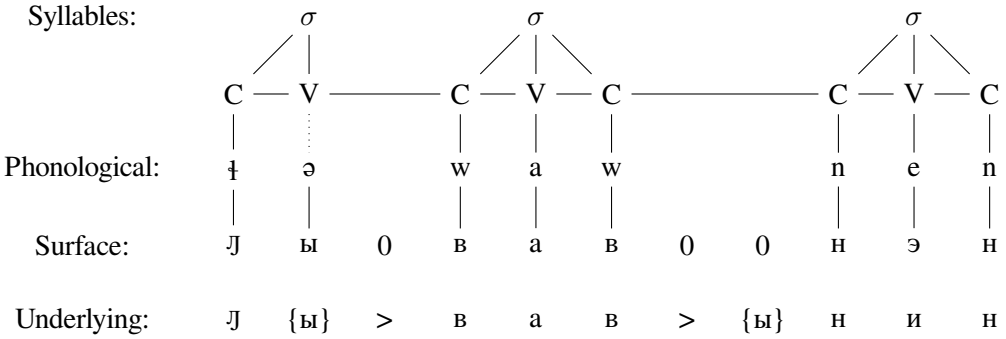

Figure 3: Process of syllabification, note that a schwa is inserted to avoid an impossible syllabic sequence *лвав-* /ɬwaw-/. Possible schwa positions are included in the `lexc` morphotactic structure using the archiphoneme {ы}. Note how in the example the first {ы} appears as ы on the surface while the second one appears as `0`, effectively being deleted. The change from *и* → *э* is as a result of the vowel harmony rule, see Figure 2.

Corpus	Tokens	Coverage	Mean ambiguity
Fairy tales (1)	26,109	76.6	1.43
Fairy tales (2)	45,654	62.2	–
Fiction (1)	29,148	58.8	–
Fiction (2)	23,352	53.1	–
Periodicals	38,552	53.7	–
Total:	162,815	60.9	1.43

Table 3: Coverage of the analyser over a range of publically-available corpora.

- verbal inflection (except for certain future paradigm cells);
- verbal derivation.

It also contains structures to parse compounds and incorporation, but turning these on causes the transducer to explode in size to the point of being uncompilable due to lack of memory. We suspect that this has to do with how we use `twol` to enforce tag placement, and intend to solve this problem going forward (see Section 7).

6 Evaluation

We performed a short evaluation on the results of the analyser for lexical coverage and also an error analysis of unknown tokens.

6.1 Coverage

The naïve coverage and mean ambiguity of the morphological analyser were determined. Naïve coverage is the percentage of surface forms in a given corpora that receive at least one morphological analysis. Forms counted by this measure may have other analyses which are not delivered by the transducer. The mean ambiguity measure was calculated as the average number of analyses returned per token in the corpus.

6.2 Error analysis

In order to determine the completeness of the implementation we performed a limited error analysis of tokens in our corpus which received no analysis from the transducer. We selected 100 forms at random and classified them into the following error classes: missing stem, incorporation, missing morphotactics, compounding, missing phonology, and typographical error.

As can be seen in Table 4 the vast majority of errors are caused by an incomplete lexicon. This is fortunately the easiest part to improve.

Surface form	Analyses
Аӈӷачормык	аӈӷачормын<n><loc>
	аӈӷы<n><side><loc>
ырытӷэй	ырыт<n><dim><sg><abs>
тэйкынин	тэйкык<v><tv><neut><aor><a_sg3><o_sg3>
,	,<cm>
ымы	ымы<adv>
	ымы<cnjadv>
рыннокон	рыннокон<n><px3sg><abs>
	рыннокон<n><sg><abs>
тэйкынин	тэйкык<v><tv><neut><aor><a_sg3><o_sg3>
,	,<cm>
ԯыгэн	ԯыгэн<part>
тэйкыпԯыткунэӈу	тэйкык<v><iv><compl><ger><ess>
	тэйкык<v><iv><compl><gna_res>
	тэйкык<v><tv><compl><ger><ess>
рырыткувнин	*рырыткувнин
.	.<sent>

Figure 4: Example of output from the analyser for the sentence *Аӈӷачормык ырытӷэй тэйкынин, ымы рыннокон тэйкынин ԯыгэн тэйкыпԯыткунэӈу рырыткувнин* 'By the coast he made a little bow, he made an arrow, just after finishing making them he broke them.' An asterisk marks a word unknown to the analyser.

Category	Frequency	Percentage (%)
Missing stem	82	75.2
Missing morphotactics	15	13.7
Incorporation	7	6.4
Missing phonology	2	1.8
Typographical error	3	2.7
Total:	109	100

Table 4: Proportion of errors by category. Note that although there were only 100 words selected, the number of errors adds up to more than 100 as some words evinced more than one kind of error.

The two words which were categorised for phonological error were *уӷӷэм-ӷэй* 'bowl-DIM' and *пыкирыԯьы-н* 'arrive-PTCP-SG.ABS'. The problem with the first one was that the stem is actually *уӷӷэмэ* and there is a process of final vowel deletion in absolutive, but this process should not happen with derivations. The problem with the second one was that the schwas behave differently with glottal stops: in some cases a glottal stop is counted as a consonant in the stem structure and therefore the schwa is inserted while in other cases the glottal stop is not a consonant, and the schwa does not get inserted. We chose the last strategy as it seemed to be more frequent. Treating this going forward may involve including rules to deal with irregularities.

Multiple words lack stems, but we can guess the form and what the stem looks like, for example, *ы'ныргы-тку-н* is clearly a verb with its subject in third plural and its object in third singular with an iterative derivation, like 'A3PL-**stem**-ITER-O3SG'.

An example of missing morphotactics is *пэтԯе-ӷэй*, which can be glossed as 'soon-DIM', although in our transducers there is no diminutive derivation for adverbs.

Typographical errors could be exemplified by *тытԯепы* 'open a door', a word that exists in our dictionaries as *тытԯёпы*, with *ё* instead of *е* (this is a common misplacement with Cyrillic alphabet).

An example of a wordform that caused both an 'incorporation error' and a 'missing stem error' is *тымэмыԯ-енаванӈаты-пԯытко-гъа-к*, which lacks a known stem, but can be glossed as 's1SG-seal-**stem**-

COMPL-TH-S1SG', with a completive derivational suffix and a thematic inflectional element. The word 'seal' is clearly incorporated in a verbal form.

7 Future work

As can be seen from the error analysis, the most pressing concern is to expand the lexicon. Unfortunately there are no other machine-readable published dictionaries of Chukchi, so to a certain extent this would need to be done by hand. One possible approach would be to use a guesser module which could guess the tags for unknown roots and then check them manually.

We are also planning to come up with a solution for incorporation and compounding and are currently investigating possible approaches.

A third idea is to look at reworking some of the phonological processes. Either by splitting all of the processes into separate rules by switching to rewrite rules, as in e.g. Chen and Schwartz (2018), or alternatively having several levels of two-level rules, e.g. splitting the application of schwa epenthesis (see Figure 3) into a separate ruleset.

As this project is now cooperating with the research community working on Amguema dialect,[8] another possible avenue for improvement would be to adapt the analyser to this variety of Chukchi.

In the longer term, we would like to investigate the creation of a spellchecker, although the coverage of the lexicon is currently too small, the sparsity of corpus data and the complexity of Chukchi morphology make finite-state spellchecking, such as described in Pirinen and Lindén (2014) a promising avenue. Making a functional tool would also give the opportunity to engage with the Chukchi-speaking language community itself, who have so far unfortunately not been directly involved in the development process.

In addition given the significant interaction between morphology and syntax in Chukchi, and the lack of any existing treebanks for polysynthetic languages, a treebank would be an interesting idea to work on.

8 Concluding remarks

We have presented the first finite-state morphological analyser for Chukchi, and the first computational analyser which can treat a large part of its verbal morphology. The analyser is free and open-source, meaning that it can be used and extended by anyone interested.[9]

Acknowledgements

We are deeply grateful to Michael Dunn and Maria Pupynina for their help in the early stages of the project and to the anonymous reviewers for their extensive and useful comments. The article was prepared within the framework of the Academic Fund Programme at the National Research University Higher School of Economics (HSE) in 2016 — 2018 (grant №17-05-0043) and by the Russian Academic Excellence Project «5-100». It was also supported through the 2017 Google Summer of Code.

A Tagset

References

Arkhangelsky, T. (2012). *Принципы Построения Морфологического Парсера Для Разноструктурных Языков*. PhD thesis, Lomonosov Moscow State University.

Arppe, A., Cox, C., Hulden, M., Lachler, J., Moshagen, S. N., Silfverberg, M., and Trosterud, T. (2017). Computational modeling of the verb in Dene languages. The case of Tsuut'ina. In *Working Papers in Athabascan Linguistics*, Red Book, pages 51–68. Alaska Native Language Center.

Chen, E. and Schwartz, L. (2018). A morphological analyzer for St. Lawrence Island / Central Siberian Yupik. In *Proceedings of the 11th Language Resources and Evaluation Conference (LREC'18)*, Miyazaki, Japan.

[8] More about this group can be found at `https://ling.hse.ru/chukchi/`

[9] The code, corpora and error analysis results described in this paper are available from GitHub at `https://github.com/BasilisAndr/chkchn` under a free/open-source licence.

Tag	Description	Tag	Description	Tag	Description	Tag	Description
<v>	Verb	<compl>	Completive	<tv>	Transitive	<side>	Derivation 'side'
<n>	Noun	<ptcp>	Participle	<iv>	Intransitive	<a_sg1>	A agreement SG1
<adj>	Adjective	<dem>	Demonstrative	<abs>	Absolutive	<a_sg2>	A agreement SG1
<adv>	Adverb	<dim>	Diminutive	<erg>	Ergative	<a_sg3>	A agreement SG1
<cnjadv>	Adverbial conjunction	<ger>	Gerund	<loc>	Locative	<o_sg1>	O agreement SG1
<cnjcoo>	Co-ordinating conjunction	<gna_res>	Resultative	<ess>	Essive	<o_sg3>	O agreement SG3
<post>	Postposition	<intn>	Intentional	<sg>	Singular	<o_pl2>	O agreement PL2
<prn>	Pronoun	<neut>	Neutral	<pl>	Plural	<o_pl3>	O agreement PL3
<ij>	Interjection	<pers>	Personal	<aor>	Aorist	<s_sg1>	S agreement SG1
<part>	Particle	<px3sg>	Possessive SG3	<caus>	Causative	<s_pl3>	S agreement PL3

Table 5: List of all tags used in the article.

Dunn, M. J. (1999). *A grammar of Chukchi*. PhD thesis, The Australian National University. `http://hdl.handle.net/1885/10769`.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Hulden, M. (2009). Foma: a finite-state compiler and library. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, pages 29–32. Association for Computational Linguistics.

Hulden, M. and Bischoff, S. T. (2008). An Experiment in Computational Parsing of the Navajo Verb. In Hulden, M. and Bischoff, S. T., editors, *Coyote Papers*, volume 16, pages 110–118. University of Arizona Linguistics Circle.

Karttunen, L. (1993). *The Last Phonological Rule: Reflections on constraints and derivations*, chapter Finite-state constraints. University of Chicago Press.

Kazeminejad, G., Cowell, A., and Hulden, M. (2017). Creating lexical resources for polysynthetic languages–the case of Arapaho. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 10–18.

Linden, K., Silfverberg, M., Axelson, E., Hardwick, S., and Pirinen, T. (2011). *HFST–Framework for Compiling and Applying Morphologies*, volume 100 of *Communications in Computer and Information Science*, pages 67–85.

Moshagen, S., Trosterud, T., Rueter, J., Tyers, F. M., and Pirinen, T. A. (2014). Open-source infrastructures for collaborative work on under-resourced languages. In *Proceedings of CCURL workshop 2014 organised with LREC2014*.

Pirinen, T. A. and Lindén, K. (2014). State-of-the-art in weighted finite-state spell-checking. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 8404*, CICLing 2014, pages 519–532, Berlin, Heidelberg. Springer-Verlag.

Rios, A. (2016). A basic language technology toolkit for Quechua. *Procesamiento del Lenguaje Natural*, (56):91–94.

Skorik, P. Y. (1977). *Грамматика чукотского языка: Глагол, наречие, служебные слова. Часть вторая*. Академия Наук СССР, Институт Языкознания.