Towards a Hän morphological transducer

Maura O'Leary[1], Joseph Lukner[2], Finn Verdonk[2], Willem de Reuse[3], Jonathan Washington[2]
[1]Western Washington University
[2]Swarthmore College
[3]The Language Conservancy
Correspondence: Maura.O'Leary@wwu.edu, Jonathan.Washington@swarthmore.edu

Abstract

This paper presents work towards a morphological transducer for Hän, a Dene language spoken in Alaska and the Yukon Territory. We present the implementation of several complex morphological features of Dene languages into a morphological transducer, an evaluation of the transducer on corpus data, and a discussion of the future uses of such a transducer towards Hän revitalization efforts.

1 Introduction

In this paper, we present work towards a morphological transducer for the Dene language Hän. The paper provides background on Hän, data collection, and morphological transducers (§2); overviews decisions made during implementation as well as our approaches to various challenges presented by Hän morphology (§3); and offers a preliminary evaluation (§4), directions for future work (§5), and some concluding thoughts (§6). The eventual goal is for this transducer to complement ongoing revitalization efforts.

2 Background

2.1 Hän

Hän (ISO 639-3: `haa`) is a Dene (more specifically, Northern Athabaskan) language spoken in the Native Village of Eagle in Alaska, USA, and in Moosehide, Yukon Territory, Canada. Hän is a critically endangered language, with only five remaining native speakers. While the number of native speakers is low, the communities in both Eagle and Moosehide are both engaged in significant revitalization efforts, including locally taught introductory language courses, language teacher training, and the creation of learning materials (lessons, textbooks, flashcards, etc.).

The primary complication in the process of learning (and thereby also in the process of revitalizing) an Athabaskan language is the rather complex verbal morphology. Verbs often surface with a string of both derivational and inflectional prefixes, which can be difficult for speakers of less-inflecting languages such as English. The complexity of Hän verbs stands in stark contrast to every other lexical category, which are at most bimorphemic.

In order to progress the community's revitalization efforts, there is a clear need for an understanding of Hän's verbal morphology. Understanding the inner workings of verbs has been a long-standing battle for many Athabaskan languages (see Rice, 2000, for an overview of many of the relevant works), and Hän is no exception. We intend for this transducer, and the resources which stem from it, to clarify the inner workings of the Hän verb as an aid to future Hän language learners.

Table 1 presents the structure of verbs in Hän, with some example verb forms broken down accordingly in Table 2. Each cell represents a distinct morpheme "slot". Many of the slots are optional—a valid verb form must contain at a minimum a stem marked for aspect and a subject marker. However, some verbs additionally require other elements, such as a theme or disjunctive prefix. Additionally, several slots interact with one another; for example, generally subject morphology is indicated in the slot before the stem, but plurality is indicated by a morpheme's occurrence in the "plural subject" slot for 3rd person plural and another morpheme's occurrence in the "deictic subject" slot for 1st person plural. Object marking and the presence of a reflexive morpheme appear to be mutually exclusive. 3rd person singular object markers vary depending on the person features of the subject (Lehman and O'Leary, 2019). Object marking is used only if an overt object DP is not present in situ in the verb phrase (Manker, 2014). Additionally, the specific form of subject marking depends on the classifier (l, ɬ, 0, or d), the aspect (imperfective, perfective, etc.), and the conjugation marking (0, dh, gh) associated with the verb stem (de Reuse and Las, 2014). Verb stems alternate irregularly for a given lexeme based on aspect and sometimes number of the subject (de Reuse, 2015b,a).

10	9	8	7	6	5	4	3	2	1	0
(disjunctive prefix)	(pl. subj.)	(object)	(deictic subj.)	(reflexive)	(directive)	(future/ inceptive)	(gender/ qualifier)	(theme)	conjugation marker, subject, classifier	stem

Table 1: The structure of verbs in Hän, with numbers assigned to each prefix slot. The stem occurs at the end of a verb form, with prefixes stacking before it. Prefix slots that are not used in every verb form are described in ()s.

	10	9	8	7	6	5	4	3	2	1	0
a.	nä- ITER-		n- OBJ.2SG-			u- DIR-			n- THM-	ök- (ɬ/0) SUBJ.3SG-	gòt punch.IMPF
b.		hë- SUBJ.PL-						jë- GENDER-	n- THM-	èh- (ɬ/dh) SUBJ.3-	tlot boil.PERF

Table 2: Morphological breakdown of two example verb forms: (a) *nänunökgòt* 'I keep hitting you (sg) over and over again' and (b) *hëjënèhtlot* 'they boiled (a liquid)'. Numbers corespond to those for prefix slots in Table 1. Classifier and conjugation marker are specified in the gloss of the subject prefix (slot 1).

2.2 Language data and elicitation

The data used in this project comes primarily from in-person elicitation done by the fourth author between 2006 and 2012 (de Reuse, 2015b) and, to a lesser extent, from in-person elicitation done by the first author between 2016 and 2022. (As is discussed in §4.1, short stories written by one of the speakers are also used to test coverage and build the lexicon.) In addition to descriptive fieldwork, both the first and fourth authors have also been involved directly in Hän revitalization efforts since 2017, with projects yielding in-person language workshops and physical language learning materials (flashcards, language games, a phrasebook, and a short textbook). As revitalization efforts continue, the first author remains in close contact with the Eagle Village Chief—who is also daughter to one of the remaining speakers and niece to two others—so that all efforts can be made to fit the desires and needs of the language learning community. §5 discusses the potential future uses of a Hän morphological transducer in the revitalization process, which the community has shown great excitement for.

2.3 Finite-state transducers

A morphological transducer is a finite-state model of a language's morphology such that valid forms of a language receive one or more analyses (morphological analysis), and a valid form of a language is output when an analysis is input (morphological generation), as illustrated in Figure 1.

A finite-state transducer can be a useful tool for a marginalized language for a number of reasons. Most directly, it can be used for linguistic research and analysis of texts. It can also expand access to language

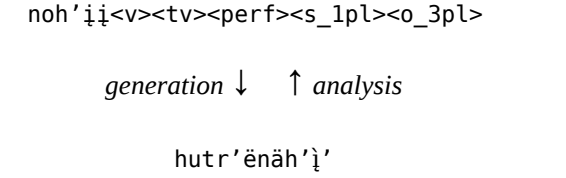

Figure 1: An example of morphological analysis and generation, as different directions in the mapping between an analysis (`noh'ɨɨ<v><tv><perf><s_1pl><o_3pl>`) and a form (`hutr'ënäh'ɨ'`). The example translates roughly as 'We saw them.'

technology, a crucial element for vitality of a language in the 21st century (Kornai, 2013), including as a core component of tools such as machine translation systems and spell checkers (Khanna et al., 2021). Additionally, a finite-state transducer can be useful for language revitalization as a component of pedagogical tools, such as Computer-Assisted Language Learning tools (Snoek et al., 2014; Katinskaia et al., 2018; Ivanova et al., 2019), word-form creators (Fernald et al., 2016; Kazantseva et al., 2018), and paradigm generators.[1] It is our intention to move to integrating the present transducer into any of these tools that the Hän community might find useful once the transducer is mature enough.

3 Implementation

One major challenge presented by Dene languages for development of a morphological transducer is the fact that the verb morphology is complex (§2.1) and almost entirely prefixational. Morphological analy-

[1] A prototype paradigm generator using transducers is available at https://apertium.github.io/apertium-paradigmatrix/ with source code at https://github.com/apertium/apertium-paradigmatrix.

ses of the type returned by transducers are usually organised in a suffixational order: `lemma`, `POS tag`, `subcategory tag(s)`, `grammatical tag(s)`, e.g., `noh'ɨɨ<v><tv><perf><s_1pl><o_3pl>`, where `noh'ɨɨ` is the lemma, `<v>` represents the part of speech (verb), `<tv>` represents the subcategory (transitive verb), and `<perf><s_1pl><o_3pl>` constitute grammatical tags (perfective aspect, first-person plural subject, third-person plural object). This order is much easier to implement when subsequent grammatical tags match the order of added [suffixational] morphology and occur after the stem; formalisms that rely on continuation lexicons, such as `lexc`, fail to offer a straightforward solution for non-suffixational morphology. For such languages, including Dene languages, a combination of several approaches is used to circumvent these limitations: the use of flag diacritics, intricate continuation lexicons, and collapsing intricate verbal morphology into simplified "zones" (Harrigan et al., 2017; Arppe et al., 2017; Holden et al., 2022). The main disadvantages of these approaches seem to be cleanliness of code (and hence maintainability) as well as transducer size and compilation and runtime speed.

To get around these limitations of previous approaches, the `lexd` formalism and compiler (Swanson and Howell, 2021) was used to implement a model of Hän morphology. The `lexd` formalism was designed to handle non-suffixational morphology efficiently, and has proven effective for other languages which make use of non-suffixational morphology (Washington et al., 2021; Christopherson, 2023).

We use the Apertium framework (Forcada et al., 2011; Khanna et al., 2021) for compilation scripts and other features and HFST format and tools (Linden et al., 2011) for storing and working with the compiled transducer, and adhere closely to the Apertium tagset standards.[2]

The remainder of this section reports on the implementation of the lexicon (§3.1), aspectual verb stem alternations (§3.2), and distributed morphology (§3.3), as well as how we deal with spelling variation and tone spreading (§3.4), and an initial foray into implementing a guesser (§3.5).[3]

[2]Described at `https://wiki.apertium.org/wiki/List_of_symbols`.

[3]Source code is available under a free/open license at `https://github.com/SwatLangTech/apertium-haa/`. All reports of code and performance are based on the latest code at time of submission: revision `b334130`, dated 2025-01-17.

3.1 Lexicon

We have mostly focused our efforts on implementing verbal morphology. Other parts of speech have been included in the lexicon to "clear out" the list of top unanalysed forms over corpora so that verb forms become more visible for additional morphology work. A first stab at non-verbal morphology, which is limited in Hän to pronominal possessor prefixes on nouns and pronominal prefixes on prepositions denoting indirect objects, has been implemented. The number of stems of various types are listed in Table 3.

part of speech	unique	total
nouns	167	183
verbs	15	64
adjectives	18	20
prepositions	15	17
adverbs	6	8
conjunctions	3	4
modal words, determiners, pronouns, numerals, anthroponyms, etc.	22	23
total	246	319

Table 3: The number of stems of various parts of speech: unique excludes spelling variants or context-dependent stems; total is the total number of entries in each lexicon.

Uninflected verb stems in Hän are never uttered in isolation, and verbs have different stems depending on their patterning with aspectual morphology, so verb lemmas must inherently be inflected for subject and aspect. We originally selected the 3rd person singular imperfective form of a verb as its lemma for morphological reasons—primarily that this form is also used as a base on which the 1st person plural and 3rd person plural forms are built, and thus is present in three of six person/number combinations. However, recent speaker judgments suggest that the 1st person singular imperfective form feels like a more appropriate label for the verb, so we will be transitioning to a 1st person singular imperfective lemma system.

3.2 Aspectual verb stem alternations

Verb stems in Hän take different forms depending on the aspect marker they pattern with, as well as (in some cases) whether the subject is singular or plural. Since these alternations are unpredictable, they could not easily be encoded as phonological alternations. Instead, we implemented these alternations using filter tags, a feature of `lexd`. An example is provided in

Code Block 1.

Additionally, subject markers take different forms based on the classifier and conjugation marker associated with the verb. These alternations are also unpredictable and could not be treated as phonology. In this case also we used filter tags to match verb entries to the appropriate set of subject markers.

The result is that there are currently 173 entries in the subject lexicon (excerpt in Code Block 2), which includes the morphology for all person categories matched to each combination of classifier and conjugation marker, as well as variant forms.

Besides indexing the relevant tags in each entry of each lexicon, tags must be matched at the level of the pattern (pattern example with tags shown in Code Block 3).

3.3 Distributed morphology

The implementation of the transducer needed to model the distribution of subject morphology across three slots of the verb structure (Table 1). This was done by making multi-column lexicons for verb morphology, as shown in Code Block 1. The verb lexicon currently includes four columns: one for disjunct prefixes associated with the given verb, one for the directive prefix associated with the given verb, one for the theme prefix associated with the given verb, and one for the stem. This treats the lexical entries for verbs as consisting of all four parts.

The different columns and associated morphology (e.g., Code Block 2) are referenced from a pattern that follows the structure of verbs in Hän. The pattern for transitive verbs currently in the transducer is shown in Code Block 3. This pattern does not yet implement the disjunctive prefix, or the gender/qualifier slot.

3.4 Spelling variation and tone spreading

There are a number of challenges for analysis related to orthography.

First of all, due to the small number of remaining speakers, as well as inconsistencies among our data sources, tokens of the same word often vary in spelling. We add variant forms of an entry to the lexicon in a way where only one variant (the one determined to be canonical) is included in the generator, but all variants are included in the analyser. This is done by simply including a control sequence (`Dir/LR`, a convention established by the Apertium community) in the comments of all but the canonical form in the `lexd` file, and including code in our compile script to strip all lines containing that control sequence when compiling the generator. Currently there are 56 instances of this control sequence in the transducer code.

Additionally, the tone system of Hän features interlexical tone spreading: if the last (or only) syllable of a word has a low tone, this low tone can spread to the following syllable of a subsequent word, if that syllable is not then followed by another low tone (Lehman, 2018). Notably, this spreading skips over schwas. In many instances however, this standard is not strictly adhered to in the orthography. Practically, this means that the first non-schwa vowel of a token may be written with an otherwise unexpected low tone (e.g., à for expected ä).

A related challenge is the differing encodings of various characters. For example, the 'ą̀' character may be encoded as the character 'a', followed by a combining ogonek, followed by a combining diaeresis, followed by a combining grave (which we treat as the canonical encoding).[4] However, it may also be rendered with any order of combining diacritics, or with a precomposed character (such as 'ä' or 'ą') with only the additional diacritics added as combining characters (again, in any order). Normally the transducer will only recognise characters in the particular encoding that material is entered with, and not visually similar characters with different encodings.

To overcome regular spelling alternations, differing encodings, and the possibility of an additional low tone, we implemented a layer of "spellrelax" rules (which allow for alternative spellings), implemented as a list of `foma`-style rules (each its own mini transducer). Each rule allows alternate character sequences for a given canonical character sequence, and the combined ruleset is compose-intersected with the base transducer to create the final analyser. An example of two spellrelax rules is provided in Code Block 4. Currently there are 28 implemented spellrelax rules for the Hän transducer.

3.5 Guesser

By leveraging the morphological patterns of the transducer and a regular expression, a transducer may be used as a guesser. A guesser is a transducer which analyzes forms of stems which are not part of the transducer's lexicon. The output when analyzing such a

[4]The canonical order in the transducer is based on Unicode NFKD (Normalisation Form Compatibility Decomposition); we do not perform the additional composition required of NKFC (NFKD, followed by Canonical Composition) in order to maintain compatibility with the Hän keyboard available from the Yukon Native Language Centre (`https://ynlc.ca/fonts-keyboards/`), and so that the low-tone diacritic may be directly manipulated in cases of tone sandhi.

```
LEXICON VerbStem-Iv(4)

[0cl,impf,0cm,sg]:  [0cl,impf,0cm,sg]:  [0cl,impf,0cm,sg]:n>  nähaa:haa[0cl,impf,0cm,sg]
[0cl,perf,n,sg]:    [0cl,perf,n,sg]:    [0cl,perf,n,sg]:n>    nähaa:zhaa[0cl,perf,n,sg]
[0cl,fut,0cm,sg]:   [0cl,fut,0cm,sg]:   [0cl,fut,0cm,sg]:n>   nähaa:haw[0cl,fut,0cm,sg]
[0cl,perf,n,pl]:    [0cl,perf,n,pl]:    [0cl,perf,n,pl]:n>    nähaa:jeww[0cl,perf,n,pl]
[0cl,fut,0cm,pl]:   [0cl,fut,0cm,pl]:   [0cl,fut,0cm,pl]:n>   nähaa:däẁ[0cl,fut,0cm,pl]
```

Code Block 1: An example of a verb entry for the verb *nähaa* ‘go, come, arrive’. Filter tags are specified within `[]` and separated by commas, and are used to encode grammatical properties of the lines (e.g., `[0cl,impf,0cm,sg]` encodes 0-classifier, imperfect, 0-conjugation marker, singular). Columns (discussed in §3.3) are disjunct prefix (empty with this verb), directive (empty with this verb), thematic prefix (*n*-), and stem (varying by imperfective, perfective, future, as well as singular and plural). The plural imperfective stem is not in our data sources. Content outside filter tags is separated by colons: the left side contains elements of the analysis (e.g., in the last column containing the lemma, *nähaa*) and the right side contains elements of the form (e.g., the thematic prefix *n*- and the individual stems).

```
LEXICON subject(4)

[ɬ,impf,0cm,non3Ssub,sg]:         [ɬ,impf,0cm,non3Ssub,sg]:           [ɬ,impf,0cm,non3Ssub,sg]:ök>  [ɬ,impf,0cm,non3Ssub,sg]<s_1sg>:
[ɬ,impf,0cm,non3Ssub,pl]:         [ɬ,impf,0cm,non3Ssub,pl]:tr’{E}{¨}> [ɬ,impf,0cm,non3Ssub,pl]:oh>  [ɬ,impf,0cm,non3Ssub,pl]<s_1pl>:
[ɬ,impf,0cm,non3Ssub,pl]:h{E}{¨}> [ɬ,impf,0cm,non3Ssub,pl]:           [ɬ,impf,0cm,non3Ssub,pl]:oh>  [ɬ,impf,0cm,non3Ssub,pl]<s_3pl>:
```

Code Block 2: Some examples of entries in the subject lexicon. The first column provides content for plural marking in third person, the second column provides content for plural marking in first person, the third column provides remaining subject marking, and the fourth column provides the relevant morphological tags. Filter tags currently must be included in every column (a limitation of `lexd`), and in this case specify that this morphology patterns with ɬ-classifier verbs, imperfective aspect, a non-third-person-singular subject, and singular or plural subject (cf. Code Block 1).

```
(subject(1) object?(1) subject(2) object?(2) :VerbStem-Tv(2) aspect(1) VerbStem-Tv(3)
subject(3) [ :{NOV} ] VerbStem-Tv(4) [ <v><tv>: ] VerbStem-Tv(2): aspect(2) subject(4)
object?(3))[^[3Ssub,non3Ssub],^[impf,perf,incp,fut,opt],^[sg,pl],^[l,d,0cl,ɬ],^[0cm,dh,gh,n]]
```

Code Block 3: The current pattern for transitive verbs (no line breaks in the transducer entry). The elements before the verb stem (`VerbStem-Tv(4)`) reference the content of the various prefix slots. Material after the anonymous lexicon that consists of `<v><tv>` tags reference grammatical tags matching the prefixes, as well as the filter tags used to match elements of lexicons to one another.

```
.o. [ ?* [ ̨ ̈(->) ̈ ̨] ?* ]
.o. [ ?* [ i ̨(->) į ] ?* ]
```

Code Block 4: Two spellrelax rules currently used in the transducer. The `.o.` character is the compose operator, to compose each rule with the other rules. The first example allows either order of ogonek and diaeresis combining diacritics. The second rule allows a precomposed 'į' character for what is encoded in the transducer as an 'i' character followed by a combining ogonek diacritic.

form is the stem in place of the lemma, a full analysis, and information about the paradigm the form was successfully analyzed using.

Initial attempts at a guesser were implemented for some of Hän's verbal morphology by adding wildcard entries to the verb lexicon (excluded from normal compilation) with filters matching each of Hän's four verb classifiers with the zero conjugation marker. These four patterns were repeated twice, once with no thematic prefix, and once with an *n* thematic prefix, for a total of 8 entries. (Quite a few more would be needed for a complete set of entries.) An example is shown in Code Block 5.

An example of output from the guesser is shown in Code Block 6, using the example *shënähtthee* 'you all are barking at me' (the verb stem of which is not in the transducer). The returned set of analyses includes the correct one (`<GUESSER_ł_0cm_nthm>tthee<v><tv><impf><s_2pl><o_1sg>`), correctly revealing that the input token is a second-person plural subject form of an imperfective stem *tthee* of an ł-classifier verb with an *n* thematic prefix and a first-person singular object. However, other analyses are returned as well.

The guesser often returns a 3rd singular imperfective of a ∅-classifier verb. This is due to the fact that the 3rd singular imperfective subject prefix is null for ∅-classifier verbs. The entirety of the input form is then guessed as the root. Removing these extraneous analyses was done by implementing `twol` rules (Code Block 7) that restrict the possibilities for guessed roots. No verb roots in the language appear to begin with a vowel or 'h' or 'n' followed by a consonant. Additional work is needed to further restrict the options, perhaps by prioritising more complex ones using weights.

4 Evaluation

The transducer was evaluated for naïve coverage (§4.2) using available texts and elicitation sentence data (§4.1) and on its runtime and space requirements (§4.3).

4.1 Corpora

The coverage of the transducer was evaluated against several texts. The first set of texts come from two collections of short stories written by native speaker Ruth Ridley (Ridley, 1983, 2018), totaling ~3.3k tokens. The stories were manually transcribed (with some augmentation using OCR) to ensure accuracy and proper encoding.

The second set of text came from elicited sentences accompanying verb paradigms in de Reuse (2015b). Sentences were extracted using a script to filter out English, author comments, organizational codes, and Hän data that was not in sentence format. After filtering, the document contained ~11.5k words.[5]

4.2 Naïve coverage

Naïve coverage was measured as the raw percentage of tokens that were analyzed by the transducer, regardless of accuracy. Coverage numbers are shown in Table 4.

The higher coverage numbers on the stories corpus can be accounted for by several factors. First of all, the elicited sentences include a full range of verbs in the language, as opposed to handful of common and domain-specific verbs as in the stories. Additionally, the stories include common nouns, prepositions, and other uninflected parts of speech that are much less common in the sentences corpus (and which were easily included in the transducer lexicon).

Other reasons the sentences corpus has lower coverage include that (1) there was minimal punctuation in the corpus, especially since the sentences did not include sentence-final punctuation; (2) there were many words with differently encoded symbols (using private-use-area code points, presumably for a custom font) which we have not yet integrated into spellrelax; and (3) this corpus contains examples from multiple speakers and dialects, and much of the attested variation has not yet been incorporated into the transducer.

Overall, the verb paradigms were the principal source of data for implementing the transducer lexicon, so it is a good sign that it does analyze a large portion of the examples in the data. Coverage on this corpus can be increased by adding more verb stems to the lexicon (the existing morphology should be robust enough to support most cases), implementing more spellrelax rules to account for differences in encoding and orthography, and including more phonolog-

[5] There are ~4.5k sentences; i.e., they are on average very short.

```
[ɬ,0cm]:   [ɬ,0cm]<GUESSER_ɬ_0cm_nthm>:    [ɬ,0cm]:n>    /([a-z'\̃\̃])+/[ɬ,0cm]
[l,0cm]:   [l,0cm]<GUESSER_l_0cm_nthm>:    [l,0cm]:n>    /([a-z'\̃\̃])+/[l,0cm]
```

Code Block 5: Two guesser entries in the verb lexicon: one for l-classifier verbs and one for ɬ-classifier verbs. The columns match those in Code Block 1; an *n* thematic prefix is included in the third column. The regular expression in the fourth column occupies both sides of the separator, so the transducer includes the matching stem on both the analysis and the morphological side. For this reason, the guesser tag must be included in a different column than (and hence occurs before) the stem.

```
<GUESSER_0cl_0cm>nähtthee<v><tv><impf><s_3sg><o_1sg>
/<GUESSER_0cl_0cm_nthm>tthee<v><tv><impf><s_2pl><o_1sg>
/<GUESSER_d_0cm_nthm>tthee<v><tv><impf><s_2pl><o_1sg>
/<GUESSER_ɬ_0cm_nthm>tthee<v><tv><impf><s_2pl><o_1sg>
```

Code Block 6: Analyses returned by the guesser given the input *shënähtthee* 'you all are barking at me' (a verb form whose stem is not in the transducer). The correct analysis is the fourth one, highlighted in bold for presentation purposes.

```
"restrict guessed forms with vowel-initial stems"
Vowel:Vowel /<= %{NOV%}: _ ;

"no hC- or nC- initial stems guessed by guesser"
C1:C1 /<= %{NOV%}: _ Cons:Cons ;
         where C1 in ( h n ) ;
```

Code Block 7: `twol` rules that restrict guesser possibilities. `{NOV}` ("no vowel") is a control symbol included in the transitive verb pattern before the stem (Code Block 3). The `/<=` operator excludes from the compiled transducer any path matching the pattern.

corpus	tokens	ambiguity	coverage
stories	3 275	1.08	60.40%
elicitation data	11 479	1.10	21.87%

Table 4: Naïve coverage results by corpus. Corpus size is presented in number of tokens as determined by the analyser. Ambiguity (average number of analyses per form) is also included.

ical rules to better predict the morphophonology of long sequences of prefixes.

4.3 Size and speed

As of publication, the generator has 19 824 states and 23 105 arcs and a non-cyclical expansion of the generator[6] yields 4 286 analysis-form pairs, taking approximately 280ms to expand on a 3.5GHz Intel i9-9900X CPU, and running a simple coverage script on the 3.3k-token stories corpus takes approximately 125ms.[7] The compiled generator is 367kB the compiled analyser is 859kB and the compiled guesser is 6.7MB

Compilation of the entire transducer—including morphology, morphophonology, guesser, and spellrelax—using a single thread on a 10-core 3.5GHz Intel Core i9-9900X CPU takes approximately 30 seconds total and uses a maximum of 652MB of RAM. Use of additional threads brings compile time down to around 14 seconds.

While these are encouraging numbers given the complexity of the existing morphology, it is difficult to know how size and speed will scale as the lexicon is expanded and additional morphology is added.

5 Next Steps

The most pressing next steps are to continue to expand the transducer in all ways, including lexicon, morphology, and phonological alternations.

The primary motivation for creating this transducer is pedagogical. Specifically, we envision the transducer's use in tools that can be used by language learners, such as a verb-form generator, a paradigm generator, or a translator working at the sentence level rather than the word level (examples for other languages cited in §2.3). Such resources would be incredibly valuable to Hän language learners, many of whom do not have the opportunities for frequent contact with the few remaining speakers. Existing revitalization materials, being limited to slide shows and printed physical materials, do not cover many verbs or full conjugation paradigms. Hence any of these resources would be a significant addition to current revitalisation efforts, but would have to be built for use by non-technical audiences (e.g., avoiding linguistic terminology wherever possible). Community leaders have expressed excitement at the prospect of materials like these becoming available to the community.

As with all resources created for Hän, prototypes will be presented to the Hän community to allow their

[6] `hfst-expand -c0 haa.autogen.hfst`

[7] All data files and utilities are stored on a 2019-era Samsung 970 Pro NVMe SSD.

preferences to guide resource development, so that the resulting resources are only those that are deemed beneficial by the speakers and learners themselves.

Finally, we also plan to account for systematic spelling and vocabulary differences found between the the Eagle (Alaska) and Moosehide (Yukon) dialects of Hän, so that any pedagogical resources produced will be equally accessible to both communities.

6 Conclusion

To our knowledge, we are publishing the first morphological transducer for a Dene language written in `lexd`. Not only have we shown that it is possible to implement Dene morphology in `lexd`, but that it has many advantages over previous approaches to Dene morphology using `lexc` (see §3): the code is much cleaner (and hence the transducer is more easily maintained and expanded), and the resulting transducer is small and its compilation and runtime speeds are fast. Our hope is that an efficient transducer will allow us to create helpful and easy-to-use language resources to aid the revitalization of the Hän language.

Acknowledgments

Thanks first and foremost go to the speakers of Hän who have shared their language with us over the past 20 years, including many who are no longer with us. Speakers involved in the data used in this project are, alphabetically: Angie Joseph-Rear, Adeline (Juneby) Potts, Archie Roberts, Bertha Ulvi, Charlie Silas, Charlie Stevens, Danny David, Doris Roberts, Edith Josie, Edward Roberts, Ethel Beck, Eliza Malcolm, Geoffrey O'Grady, Harry David, Jr., Isaac Juneby, Joseph Susie Joseph, Louise Paul, Matthew Malcolm, Percy Henry, Richard Nukon, Richard Silas, Ruth Ridley, Sarah Malcolm, Stanley Roberts, Susie Paul, Timothy Malcolm, and Willie Juneby. We also thank linguists whose work with Hän has helped us at various stages, including Michael Krauss, Jordan Lachler, Blake Lehman, Gordon Marsh, John Ritter, and David Shinen, as well as community members Georgette McLeod and Eagle Chief Karma Ulvi, who have been heavily involved in the revitalization of Hän. We also recognize work done by research assistants Ryan Baldwin and T Sallie to expand the lexicon of the transducer, and thank the ComputEL-8 organizers and four anonymous reviewers for their feedback.

References

Antti Arppe, Christopher Cox, Mans Hulden, Jordan Lachler, Sjur N. Moshagen, Miikka Silfverberg, and Trond Trosterud. 2017. Computational modeling of verbs in Dene languages: The case of Tsuut'ina. *Working Papers in Athabascan Linguistics*.

Cody Scott Christopherson. 2023. A finite-state morphological analyzer for Q'eqchi' using Helsinki Finite-State Technology (HFST) and the Giellatekno infrastructure. Master's thesis, Brigham Young University.

Willem de Reuse. 2015a. A guide to the Hän verb paradigms. Unpublished ms.

Willem de Reuse. 2015b. Hän Athabascan verb paradigms. Unpublished ms. Fairbanks: Alaska Native Language Archives, University of Alaska.

Willem de Reuse and Kathy Joy Las. 2014. Han verb prefix paradigms: digitized and rechecked version of field notes from Jeff Leer (1974, 1977, 1980) and Leer and Ridley (1982). Unpublished ms.

Theodore B. Fernald, Nabil Kashyap, and Jeremy Fahringer. 2016. Navajo verb generator. Development version.

Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Atticus G. Harrigan, Katherine Schmirler, Antti Arppe, Lene Antonsen, Trond Trosterud, and Arok Wolvengrey. 2017. Learning from the computational modelling of Plains Cree verbs. *Morphology*, 27:565–598.

Joshua Holden, Christopher Cox, and Antti Arppe. 2022. An expanded finite-state transducer for Tsuut'ina verbs. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5143–5152, Marselha, França. European Language Resources Association.

Sardana Ivanova, Anisia Katinskaia, and Roman Yangarber. 2019. Tools for supporting language learning for Sakha. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 155–163, Turku, Finland. Linköping University Electronic Press.

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2018. Revita: a Language-learning Platform at the Intersection of ITS and CALL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Anna Kazantseva, Owennatekha Brian Maracle, Ronkwe'tiyóhstha Josiah Maracle, and Aidan Pine. 2018. Kawennón:nis: the wordmaker for Kanyen'kéha. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 53–64, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tanmai Khanna, Jonathan N. Washington, Francis M. Tyers, Sevilay Bayatlı, Daniel G. Swanson, Tommi A. Pirinen, Irene Tang, and Hèctor Alòs i Font. 2021. Recent advances in Apertium, a free/open-source rule-based machine translation platform for low-resource languages. *Machine translation*, 35:475–502.

András Kornai. 2013. Digital language death. *PLoS ONE*, 8.

Blake Lehman. 2018. Tone-prominence interaction in Hän. Master's thesis, University of California, Los Angeles.

Blake Lehman and Maura O'Leary. 2019. Unexpected Athabaskan pronouns. *UCLA Working Papers: Schuhschrift: Papers in Honor of Russell Schuh*, pages 122–137.

Krister Linden, Miikka Silfverberg, Erik Axelson, Sam Hardwick, and Tommi Pirinen. 2011. *HFST—Framework for Compiling and Applying Morphologies*, volume 100 of *Communications in Computer and Information Science*, pages 67–85. Springer.

Jonathan Manker. 2014. The syntax of sluicing in Hän. In *Proceedings of the 2012 Athabascan Languages Conference.*, Fairbanks, AK: Alaska Native Language Center.

Keren Rice. 2000. *Morpheme Order and Semantic Scope: Word Formation in the Athapaskan Verb*. Cambridge University Press, Cambridge.

Ruth Ridley. 1983. *Eagle Han Huch'inn Hòdök (Stories in Eagle Han Huch'inn)*. Alaska Native Language Center, University of Fairbanks, Fairbanks, Alaska.

Ruth Ridley. 2018. Hän children's stories. Unpublished manuscript.

Conor Snoek, Dorothy Thunder, Kaidi Lõo, Antti Arppe, Jordan Lachler, Sjur Moshagen, and Trond Trosterud. 2014. Modeling the noun morphology of Plains Cree. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 34–42, Baltimore, Maryland, USA. Association for Computational Linguistics.

Daniel Swanson and Nick Howell. 2021. Lexd: A finite-state lexicon compiler for non-suffixational morphologies. In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*, pages 133–146.

Jonathan Washington, Felipe Lopez, and Brook Lillehaugen. 2021. Towards a morphological transducer and orthography converter for Western Tlacolula Valley Zapotec. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 185–193, Online. Association for Computational Linguistics.