Interactive Word Completion for Plains Cree

William Lane
Northern Institute
Charles Darwin University

Atticus Harrigan
Alberta Language
Technology Lab
University of Alberta

Antti Arppe
Alberta Language
Technology Lab
University of Alberta

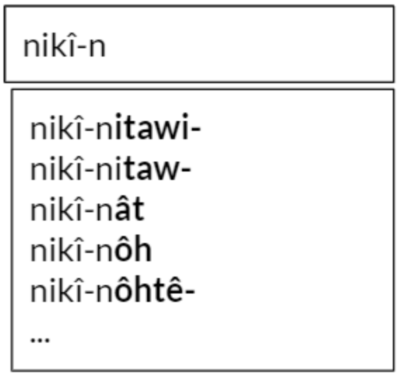
Abstract

The composition of richly-inflected words in morphologically complex languages can be a challenge for language learners developing literacy. Accordingly, Lane and Bird (2020) proposed a finite state approach which maps prefixes in a language to a set of possible completions up to the next morpheme boundary, for the incremental building of complex words. In this work, we develop an approach to morph-based auto-completion based on a finite state morphological analyzer of Plains Cree (*nêhiyawêwin*), showing the portability of the concept to a much larger, more complete morphological transducer. Additionally, we propose and compare various novel ranking strategies on the morph auto-complete output. The best weighting scheme ranks the target completion in the top 10 results in 64.9% of queries, and in the top 50 in 73.9% of queries.

1 Introduction

The ACL 2022 theme track asks how we can scale up current NLP technologies for the rich diversity of human languages. We contend that impactful work, intended to support local language community goals, does not necessarily focus on scale or current NLP technologies. Community values and motivations regarding language technology is as rich and varied as human language itself, and solutions which are well received in one community may not be adequate or appropriate in another. Context must be considered, from which novel tasks take shape.

Lane and Bird (2020)'s re-imagining of word completion for morphologically-rich languages is an example of work born from a local context. Based in Kunwinjku-speaking communities in northern Australia, we wanted to support people's desire to learn literacy and practice building the language's long polysynthetic words. This led to the idea for a tool that helps users incrementally build complex words by suggesting completions up to the next morph boundary (Figure 1). While Plains Cree and Kunwinjku speaking communities are thousands of miles apart, the authors of this work noticed some similarities in their respective contexts: both Plains Cree and Kunwinjku are polysynthetic languages, and both are working with communities to support language learning initiatives.



nikî-n

nikî-n**itawi-**
nikî-n**itaw-**
nikî-n**ât**
nikî-n**ôh**
nikî-n**ôhtê-**
...

Figure 1: Given some string prefix of a word in Plains Cree, Morph completion suggests continuations up to the next morpheme boundary (in bold), for interactive and incremental building of morphologically complex words.

Other aspects of our situations differ. For example, the FST morphological analyzer for Kunwinjku can be described as a field tool, developed by a single researcher over the course of a couple years, while the Plains Cree morphological analyzer has been in continuous development by a team for over 8 years. The Plains Cree model is much more extensive, robust and complete. How will the morph completion work transfer to such a large and technically refined project? How can we leverage our unique constellation of resources to adapt morph completion to suit Plains Cree? These are the question we set out to answer in this work.

This work examines the assumptions of morph-based auto-complete, and extends existing work to suit Plains Cree. Our contributions are: An imple-

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3284 - 3294
May 22-27, 2022 ©2022 Association for Computational Linguistics

mentation of morph-based completion algorithm for Plains Cree[1], a discussion of contextual similarities and differences between Kunwinjku and Plains Cree and how this affects the utility of the morph-based completion concept, and a novel ranking algorithm which enables the morph-completion concept to scale to a much larger, more extensive grammar.

2 Background

Speakers of morphologically complex languages are engaged in activities to maintain orality and support literacy. Two examples are the Plains Cree and Kunwinjku speaking communities.

Plains Cree (endonymically known as *nêhiyawêwin*, ISO 639-3: crk) is a member of the Algonquian family. It is the western-most Cree dialect, spoken by about 20,000 speakers in Alberta, Saskathewan, and northern Montana (Wolfart, 1973; Harrigan et al., 2017). Years of documentary linguistic work have produced extensive language resources in the form of grammars (Wolfart, 1973; Wolvengrey, 2011; Dahlstrom, 2014) and textbooks (Okimāsis, 2018; Ratt, 2016).

Kunwinjku (ISO 639-3:gup) is a member of the Gunwinyguan language family. It is spoken by an estimated 1,700 speakers in the west Arnhem region of northern Australia. Kunwinjku has its own documentary resources: grammars (Evans, 2003; Carroll, 1976), and a language primer (Etherington and Etherington, 1998). Despite these volumes, literacy in Kunwinjku is quite rare.

While by some standards these languages might be classified as “low-resource", the depth and abundance of descriptive linguistic work has paved the way for the development of computational models of Kunwinjku and Plains Cree morphology (Lane and Bird, 2019; Harrigan et al., 2017; Arppe et al., 2017; Schmirler et al., 2017; Snoek et al., 2014). Based on these computational models of morphology, language technologies are being developed to support the language goals of these communities: smart dictionaries and spellcheckers (Arppe et al., 2016), word-builder application (Lane and Bird, 2020), and intelligent language learning applications (Bontogon et al., 2018).

Finite State Morphology Morph completion models build on the established foundation of finite state models for morphological generation and analysis (Beesley and Karttunen, 2003). Under this formalism, it is customary to split the modelling task into two parts: the first task is to define the morphological inventory and valid transitions between morph classes, i.e. morphosyntax. The second handles any alternation that occurs at the morpho-phonological interface. Several (open-source) toolkits exist which implement these basic modeling capabilities: Foma (Hulden, 2009), HFST (Lindén et al., 2013), OpenFST (Allauzen et al., 2007), and Pyini (Gorman, 2016).

The Plains Cree morphological models are implemented with both HFST and Foma within the GiellaLT framework (Moshagen et al., 2014) and have been under active development for 8 years, and give a comprehensive treatment of noun (Snoek et al., 2014) and verb (Harrigan et al., 2017) morphology. As such, the Plains Cree model has had the opportunity to develop treatments for difficult-to-model features, such as reduplication. The Plains Cree model currently contains 21,232 stem (5,553 noun stems, 47 pronoun stems, 1,669 particles, 104 numerals, and 13,860 verb stems), derived from the lexical database underlying the bilingual Cree-English dictionary by Wolvengrey (2001).

The Kunwinjku model, on the other hand, is implemented using Foma, and has only been under periodic development for the last 2 years. In terms of size, the Kunwinjku FST contains significantly fewer stem entries: 573 verb stems[2], and 748 noun stems (Lane and Bird, 2019).

Despite these differences in implementation and scale, we show in this work that FST morph completion can be successfully adapted to work with Plains Cree.

2.1 FST-based Morph Completion

Lane and Bird (2020) present an approach to automatic word completion intended to assist language learners and speakers of morphologically complex languages who are building confidence in writing. For example, in Kunwinjku the verb stem *bawo* means “to leave”. This stem can then be inflected to convey subject, object, tense, comitative, and adverbial information:

[1] The source code for the original FST and the morph-completion model is available online at `https://github.com/giellalt/lang-crk`

[2] Though these forms can combine with derivational affixes to create a number of additional stems.

(1) bene-bad-yi-bawo-ng
3UA.3SG.P-now-COM-leave-PP
"The two of them left him with it" [E.10.162]

Building valid surface forms poses a challenge for learners of the language who may not yet have mastery of the morphology and orthography. Moreover, the vocabulary of morphologically complex languages is a combinatorial function of morpheme possibilities, making word-level prediction intractable. This use case drives the reconception of word completion as prediction up to the next morpheme boundary, to incrementally and interactively assist in the building of complex words.

The model is implemented as an extension to a standard finite state morph analyzer, and assumes that the FST model contains some intermediate representation in which morph boundaries are explicitly marked.

In brief their finite state algorithm:

1. Alters the existing morphological analyzer so that it does not remove morph boundary symbols
2. Recognizes all possible prefixes composed of user input followed by any character up to the next morph boundary symbol.
3. Generates a list of completions possible from the given prefix, constrained by the space of morphotactically valid words defined by the morph analyzer.

A detailed explanation of their algorithm, and implementation examples can be found in (Lane and Bird, 2020). They deploy their model in a Kunwinjku dictionary interface, serving a list of partial completions which are refreshed per keystroke. The user builds complex words incrementally, guided by the FST model. When a word is fully formed, the interface queries the dictionary database, using the regular morph analyzer to retrieve relevant lexical entries.

3 Adaptation for Plains Cree

Adapting the autocompletion algorithm to Plains Cree is relatively straightforward, owing in part to the similarities between Plains Cree and Kunwinjku. Like Kunwinjku, Plains Cree is a polysynthetic agglutinating language (Wolfart, 1973). Also like Kunwinjku, Plains Cree verbs have been described using templatic morphology: According to (Wolvengrey, 2012), there are 8 prefixal slots plus some amount of reduplication.

Suffixially, Wolvengrey (2012) provides 10 seperate slots, but in practice, these are regularly chunked together into a single portamanteau morph (Harrigan et al., 2017; Okimāsis, 2018). For this reason, Plains Cree is often treated as a mostly-prefixing language, similar to Kunwinjku.

Both Kunwinjku and Plains Cree also exhibit word-internal dependencies as well as noun incorporation. In terms of agreement, Kunwinjku verbs exhibit circumfixal markers for tense, where morphemes directly before and after the verb stem must agree for the feature. Plains Cree, on the other hand, exhibits dependency in its person marking (where the left-most and right-most morphemes of a verb form a circumfix) as well as its comitative derivation (where morphemes immediately to the left and right of the verb stem constitute a circumfix). Noun incorporation is present in both languages, though it is more common in Kunwinjku, where it occupies a slot in the prefixal morphology. Where Noun Incorporation is present in Plains Cree, it interrupts the verb stem itself and is rare enough to often be lexicalized as a separate verb all together.

These similarities and differences have consequences for each language's underlying FST and thus the autocompletion algorithm. Issues of long-distance dependencies are essentially handled in an identical way, through the use of flag diacritics to restrict progression through the model (Harrigan et al., 2017; Lane and Bird, 2019).

In terms of derivational morphology, the Kunwinjku FST marks derivational morpheme boundries identically to inflectional ones. With respect to the Plains Cree FST, we have explored including derivational boundaries within stems, but have left that out of the morpheme completion solution, in part because it increases complexity, size, and speed of model, and in part since making use of derivational boundaries would split stems in a manner that would require users to have an understanding of the derivational morphology of Plains Cree that most, in particualr learners, do not possess. We opt instead to pre-compile derivational stems, thus ignoring derivational boundaries in favor of providing full-stem-length suggestions to users.

4 Presentation of Plains Cree Algorithm

In this section we give a detailed overview of our implementation, with examples written in the

XFST formalism (Beesley and Karttunen, 2003).

Our first step is to capture the full lexical side of our morphological analyzer (*Words*) with morph boundaries present (a derivational boundary / is added in conjunction with the occurrence of a lexeme-internal hyphen that is not associated with an inflectional boundary, i.e. < or >):

(2)

```
define AddBoundary [[..] -> "/" ||
                    "-" _ \[ "<" | ">"]];
define CorrectWords [Words .o.
                              AddBoundary];
```

As is done in the previous work, we define FSTs which recognize morph boundaries (Bx), and everything except morph boundaries (Ax):

(3)

```
define Bx [ "<" | ">" | "/" ];
% Note: "/" denotes a derivational
% morpheme boundary
define Ax [ ? - Bx ];
```

We define an FST which defines spelling relaxation rules. Fortunately, this can be imported directly from our existing Plains Cree spelling relaxation module, with some minor additions. That FST contains rules which allow the arbitrary substitution of long and short vowels, or the deletion/insertion of sounds in particular contexts. As a simplified example:

(4)

```
define SpellRelax [ a (->) â
,, e (->) ê ,, i (->) î
,, o (->) ô ,, â (->) a
,, î (->) i ,, ô (->) o
,, [..] (->) h || Vowel _ Stop
,, h (->) 0 || Vowel _ Stop
];
```

Next, we define a series of helper FSTs. *InsertBoundary* optionally inserts morph boundaries in any context. *NextMorph* outputs everything from a given string up until the next morph boundary. *PrefixStrings* outputs all possible prefixes of a given input. *rmBoundary* removes morph boundary symbols from the given input.

(5)

```
define InsertBoundary [0 (->) Bx];
```

(6)

```
define NextMorph [?+ [ 0:Ax ]* 0:Bx];
```

(7)

```
define PrefixStrings [?* [ 0:? ]*];
```

(8)

```
define rmBoundary [Bx -> 0];
```

We compose these FSTs to form the FST which takes a string as input, and returns a list of possible completions up to the next morph boundary:

(9)

```
define MorphComplete
[InsertBoundary
.o. NextMorpheme
.o. [PrefixStrings .o.
        [CorrectWords ">"].l].u
.o. rmBoundary
] ;
```

The FST defined up to this point can be used to produce morph completions for a given input. The FST can be made tolerant of orthographic variation by composing *SpellRelax* with *MorphComplete*:

(10)

```
regex [SpellRelax .o.
           MorphComplete];
```

Up until this point, our implementation does not differ significantly from the algorithm proposed by Lane and Bird (2020), except in the definition of morph boundaries, and in the particulars of the spelling relaxation rules. However, because the Plains Cree morphological analyzer has a much larger lexical inventory than the Kunwinjku analyzer, we found the space of possible completions–particularly when allowing for orthographic variation–to be unmanageably large. In order to make use of the output of morph completion in Plains Cree, we need to extend the original algorithm to address the issue of result ranking.

4.1 Ranking Results

The Plains Cree morph completion FST can sometimes return thousands of results for a given query. The possibility of having thousands of results increases significantly when spelling relaxation rules are introduced (e.g., compare Figure 2 with Figure 3). In order to render the model usable, it is essential to enforce a ranking of the results. We tried 4 different ranking schemes and evaluated their effect on the morph completion space.

Data All 4 approaches leverage a corpus of written Plains Cree to collect frequency statistics of various subword units. We use the morphosyntactically-tagged corpus of Arppe et al. (2020), which has recently been extended with the so-called Bloomfield texts, and which includes a frequency-sorted list of tokens and their corresponding morphological analysis. This resource counts the occurrences of 33,655 unique words across a corpus of texts, including conversations, dialogues, narratives and lectures, amounting to 242,937 words total (Wolfart, 2000; Bear et al., 1992; Kā-Nīpitēhtēw, 1998; Masuskapoe, 2010; Ahenakew, 1987; Whitecalf, 1983; Minde, 1997; Bloomfield, 1930, 1934). We refer to this resource as the AWB corpus (for Freda Ahenakew, H.

Christoph Wolfart, and Leonard Bloomfield who collected and compiled the original texts) from now on in this work.

4.1.1 Prefix Weighted FST (pWFST)

The first ranking strategy we developed uses the AWB corpus to count the frequency of all possible prefixes for each word in the word list, up to and including all complete words. These counts are converted to a probability distribution by dividing by the total number of counted prefixes. We take the negative log of this probability to obtain the weight of the prefix. Lower weights correspond to more likely prefixes.

(11) $$weight(prefix) = -log\left(\frac{c(prefix)}{c(allPrefixes)}\right)$$

Unobserved prefixes are handled by obtaining a weight of 15 plus and additional weight of 1 per character after the first. This effectively places unobserved suggestions lower than any possible observed prefix in priority, and favors shorter unobserved prefixes over longer ones. The prefix weights are composed with output of the morph completion FST. Note that only the HFST compiler and its lookup utilities, `hfst-lookup` or `hfst-optimized-lookup`, support the incorporation and presentation of weights in an FST (as presented here).

(12)
```
define PrefixWeighting
          [ObservedPrefixWeights |
           UnobservedPrefixWeights];
regex [SpellRelax .o. MorphComplete]
           .o. PrefixWeighting;
```

In the evaluation we refer to this weighting scheme as pWFST.

4.1.2 Transition Weighted FST (tWFST)

A drawback of the prefix-weighting scheme is that it assigned weights to observed prefixes without considering shared transition information between morphs. This means that the resulting WFST model which stores weights for all observed prefixes, can start to reach greater than 100 megabytes in size, which may be be prohibitive for mobile deployment scenarios.

Considering this, our second weighting scheme weights transitions rather than prefixes, and results in a smaller WFST models since transitions of various prefixes can be shared. Our transition-based weighting scheme comes from Sahala et al. (2020)'s work on a finite state morphological analyzer for Babylonian[3]. The approach uses a manually disambiguated list of surface form and analysis pairs to estimate the likelihood of final analyses, represented as a sequence of transitions from internal FST states. It does this by counting transitions between states for a given form/analysis pair, and normalizing these counts into a probability distribution for each state.

If $C_s(x : y)$ denotes the counts at state s for symbol-pairs $x : y$, then the transition weight w is defined as:

$$w = \frac{C_s(x : y)}{(f_s + \sum_{z:u} C_s(z : u))}$$

In the evaluation we refer to this weighting scheme as tWFST. As with pWFST, the HFST compiler and lookup functionalities are necessary for the inclusion and presentation of weights.

4.1.3 Transformer Language Model Ranking

Language models are probability distributions over a sequence of tokens. Perplexity is a measure used to relate how well a given sequence of tokens fits a trained language model. Given a language model q, the perplexity of a sequence of tokens $t_1, ..., t_n$ is calculated as follows:

$$PP = e^{-\frac{1}{n}\sum_{i=1}^{n} ln\ q(t_i)}$$

Lower perplexity scores denote greater coherence according to the language of the training data.

For the purpose of ranking possible morph completions generated by a finite state transducer, we train a language model to represent the language of valid prefixes in Plains Cree, according to statistics gleaned from a corpus of text.

We process the corpus for the language modelling task by splitting the text into word level tokens. The list of tokens is then divided into an 80/10/10 train/validation/test split. We then split each token in the data into its set of all possible prefixes with the beginning and ending word boundaries marked. For example, the word "mîtos" becomes the set of instances:

(13) <BOS> m <EOS>
<BOS> m î <EOS>
<BOS> m î t <EOS>
<BOS> m î t o <EOS>
<BOS> m î t o s <EOS>

[3] The code for the weighting scheme, written by Mikka Silfverberg, can be found at `https://github.com/mpsilfve/fst-corpus-weights`

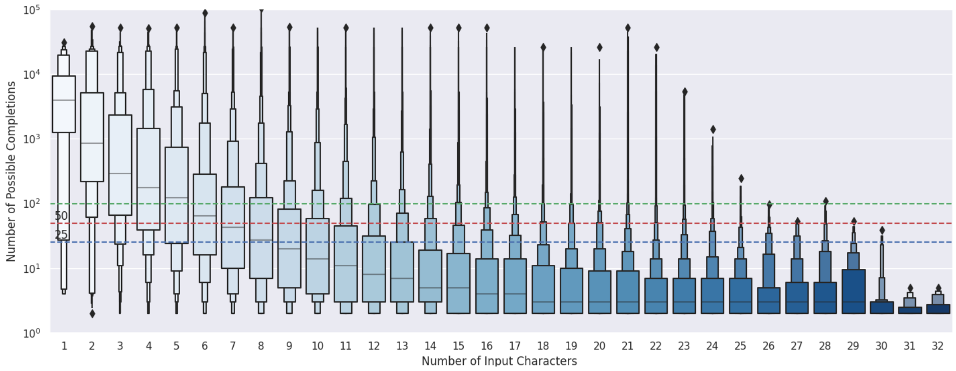


Figure 2: The distribution of completion options for Plains Cree verbs by prefix length. The prefixes are derived from a sample of 20,000 words from the Plains Cree word frequency lexicon

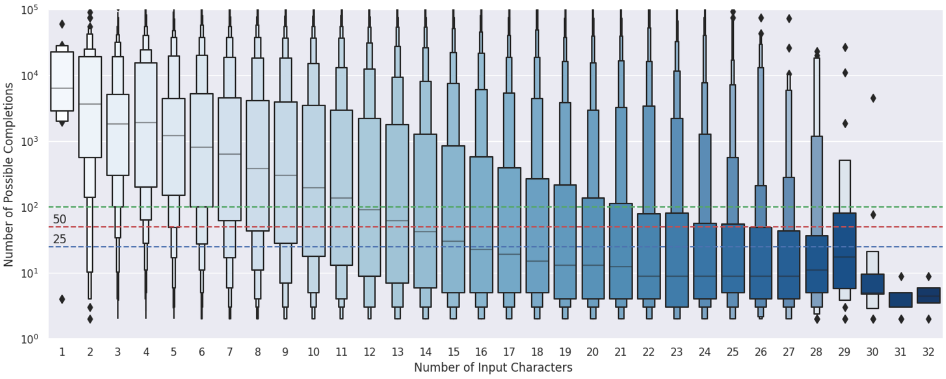


Figure 3: The distribution of completion options for Plains Cree verb with flexible spelling enabled. The prefixes are derived from a sample of 20,000 words from the Plains Cree word frequency lexicon

Preparing the data in this manner yields 1,355,740 training instances, 169,467 validation instances, and 169,467 test instances.

We train a character-based language model using Fairseq's Transformer LM architecture with default parameters (Ott et al., 2019) for 20 epochs, and select the model which minimizes error on the test set. With this model, we assign perplexity scores and rerank the output of the two WFST models. Because the number of possible results from the morph completion model can sometimes reach the order of 10^5, we limit the LM scoring to the top 500 candidates of the weighted FSTs.

5 Evaluating Morph Completion and Ranking

The intention is that this model can be deployed to support text entry in a morphologically complex language. We therefore want to measure how often the model is able to deliver a useful set of results. We define a "useful" result set as one which returns at least one completed prefix which is a proper substring of the target full-word, within the top N ranked results. For example, if $N = 10$, our target word is *nikî-kitêyimikawinân* and our query is *ni*, then a result of *nikî-* appearing in the top 10 results is counted as useful.

In order to evaluate the usefulness of the models systematically, we randomly sampled 100 unique fully-inflected word forms from the "Dog Biscuits" story by Solomon Ratt which is publicly available online.[4] These words are broken down into their complete set of prefixes, which created a set of 1,322 prefixes. Each prefix is used as a query, retrieving result sets at $N = 10, 25, 50$, and we report on the percent of queries which return a valid completion in the top N for each of the ranking strategies described in Section 4.1 (See Table 4 for results).

6 Results

We measure the completion space of two versions of the morph completion model. Given a large sample of model inputs (prefixes), the x-axis repre-

[4] https://creeliteracy.org/2014/01/20/dog-biscuits-y-dialect-with-audio/

	Top 10	Top 20	Top 50
FST	38.4	42.7	48.0
pWFST	64.9	69.5	73.9
tWFST	40.1	47.6	61.2
pWFST.LM	40.3	48.0	60.2
tWFST.LM	48.7	55.6	66.3

Figure 4: Given a random sample of 1,322 prefixes derived from 100 Plains Cree verbs, we show the proportion of these prefixes which which produce a valid completion in in the top N ranked results.

sents all prefixes of length x. The y-axis shows the number of completions generated from the model, and the data is represented as distributions over all inputs of length x. The first model shows the completion space of the sample when we strictly adhere to orthographic standard (Figure 2). The second model implements spelling relaxation (Figure 3). The effect of spelling relaxation on the number of possible completions is, as one would expect, a significant upward shift in the number of possible completions across all character positions.

Given that the purpose of morph-based completion is to help guide the user to build out complex words, we would prefer to deploy a spelling-relaxed version of the model. However, the magnitude of the measured completion space of this model would make this infeasible, as the median number of completions stays above 100 up until 12 characters of the input have been typed. Thus, coming up with an effective weighting strategy is absolutely essential in order to have a model that can handle orthographic variation in the input.

The baseline, unweighted strategy can be seen in Figure 5. Here, the distribution of rankings roughly imitates the shape of the full completion space (Figure 3), with the majority of mass occurring above our ideal ranking threshold of 10. To be precise, with the baseline no-ranking strategy, 38.4% of sampled queries result in a valid completion ranked in the top 10 results, 42.7% give a valid completion in the top 20, and 48.0% give a valid completion in the top 50.

In contrast, the best weighting scheme is the WFST, which significantly improves the distribution of rankings compared to the baseline, with the majority of queries providing valid completions in the top 10. More precisely, 64.9% of prefix queries result in a a valid completion ranked in the top 10, 69.5% in the top 20, and 73.9% in the top 50 (Figure 6).

7 Qualitative Evaluation and Discussion

This section gives an overview of the use of the morphological autocomplete system by one of the authors, an English native, second language learner of Plains Cree.[5] This use case aligns with a major subset of potential users: literate but non-fluent learners of the language.[6] By restricting the autocomplete results to the top 10 most heavily weighted items, we have found the system to perform quite well. The system was evaluated by typing the basic introductory phrase *tânisi nitôhtemak! Atticus nitisiyihkâson êkwa kêkâ-nistomitanaw ê-tânitahtopiponêyân. nitatoskân amiskwaciy-wâskahikanihk*, which translates to "Hello friends! my name is Atticus and I am 29 years old. I work in Alberta." This phrase was chosen as it contains fairly common lexical items while also being a realistic use case. In typing these words, no diacritics were used, as typing a circumflex on a North American keyboard requires a number of extra strokes, and such diacritics are often not included in non-professional Plains Cree writing. Additionally, *Atticus*, was not typed into the autocomplete system as it is not a Plains Cree word.

Writing this excerpt reveals interesting user experience data. While most words had the appropriate autocomplete suggestion, the word *nitisiyihkâson* could only be suggested by typing all but the last two letters: *nitisiyihkâs*. Typing any less did not result in target suggestions. This was likely due to the fact that the system seemed to prefer analysing the string as beginning with the morpheme nitisiyi-, rather than the target morphemes of *ni-t-isiyihkâso-*. This is particularly notable as introductions are common, especially for language learners. Similarly, in autocompleting *kêkâ-nistomitanaw* results were unexpected. The correct breakdown for this word is *kêkâ-nisto-mitanaw*; despite this, *nisto-* and *-mitanaw* are written together orthographically. The morphological autocomplete suggestions when given the input string *keka-ni*

[5] The code for the GUI can be found at https://github.com/abbottLane/cree-wordbuilder

[6] Ideally, this evaluation would be done by a number of literate Plains Cree native speakers; however, the number of fully literate speakers who are comfortable using a computer is quite limited. Because experimenting on Plains Cree speakers requires special consideration and taking into account a low tolerance for non-canonical language (Harrigan et al., 2019), we decided to first test the system on one of the authors so as to retain the maximum number of native speaker participants for future releases.

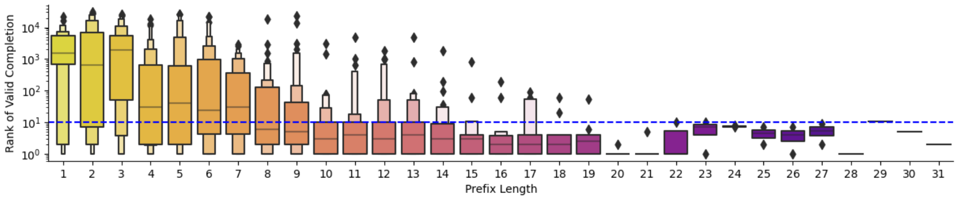


Figure 5: No ranking strategy: For 100 randomly-sampled words, we calculate all prefixes and show the distributions of ranks of all inputs per prefix length.

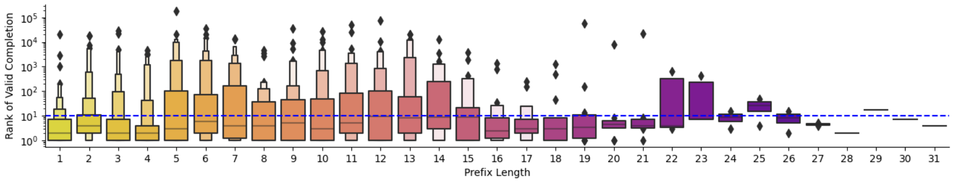


Figure 6: pWFST ranking strategy: For 100 randomly-sampled words, we calculate all prefixes and show the distributions of ranks of all inputs per prefix length.

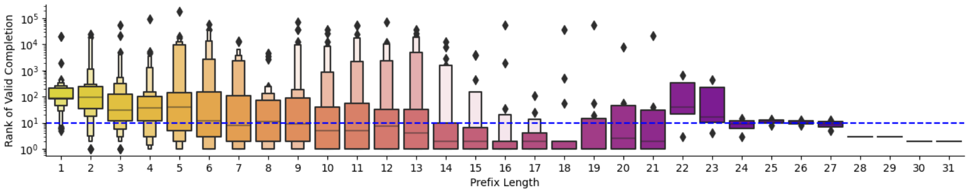
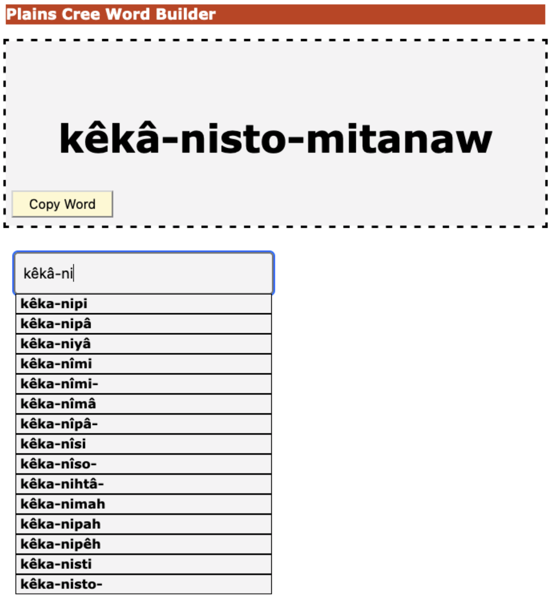


Figure 7: pWFST+LM ranking strategy: For 100 randomly-sampled words, we calculate all prefixes and show the distributions of ranks of all inputs per prefix length.



kêkâ-nisto-mitanaw

kêkâ-ni

kêka-nipi
kêka-nipâ
kêka-niyâ
kêka-nîmi
kêka-nîmi-
kêka-nîmâ
kêka-nîpâ-
kêka-nîsi
kêka-nîso-
kêka-nihtâ-
kêka-nimah
kêka-nipah
kêka-nipêh
kêka-nisti
kêka-nisto-

Figure 8: A simple GUI powered by the morph autocompletion model for Plains Cree facilitates manual exploration of the model's completion space.

produce only *kêka-nîso-*, specifically with the final hyphen. If one types *kêka-nîstom*, the system suggests *kêkâ-nistomitanaw*. The the previous two cases, the autocomplete and weighting system are working exactly as expected. The issue instead lies with the underlying corpus, which features neither *kêkâ-nistomitanaw* nor any form of the verb *nitisiyihkâson*. The corpora used as a base for weighting is mostly lectures or discussion between individuals who are otherwise familiar with each other. Unsurprisingly, this did not result in instances of individuals introducing themselves to one another or discussing anyone named Atticus.

Further, in typing only *e-* as an input string is not useful in and of itself, as all verbs can take this morpheme (written as *ê-*). In addition to the expected benefits of autocompletion, the system empowers users to type the language even if they are not entirely sure of the correct form of a word. As an example, the term *nitatoskân* comes from the root *atoskê-*. Although person marking morphology in the form of a *nit-—-n* circumfix is easy enough for learners to remember, in some conjugations, the final *ê* becomes an *â*. This is not consistent among conjugation classes, and some verb classes show the opposite alternation. As a result, second language learners can struggle with knowing whether to write *nitatoskân* or **nitatoskên*. The autocomplete system solves this problem by suggest only the correct *nitatoskâ* when the user types *nitato*. The main drawback of this system from a user perspective is that target completions were rarely the top most suggestions, but this appears to be due to minimal training data for the weighting, and is not critical as long as the user is competent enough to know which suggestions are categorically incorrect.

8 Conclusion

In this paper we presented an approach to morph-based autocompletion for Plains Cree. Informed by our particular context and availability of corpus data, we expanded on their approach by exploring three different weighting schemes to rein in the magnitude of possible completions per query, which are a result of our need to accommodate a more complex FST grammar, and greater orthographic flexibility. Our results show that all three weighting schemes go a long way to move target string rank distributions below desired thresholds, with the lexical weighting approach ranking the target completion in the top 10 results in 64.9% of queries, and in the top 50 in 73.9% of queries. The qualitative evaluation highlighted the usefulness of using an underlying FST to generate completions: long-distance dependencies and circumfixes are respected by the autocomplete algorithm, and so morphotactic integrity is preserved. Additionally, spelling relaxation rules in the underlying FST mean that the user does not need to worry as much about inputting diacritics, or exact spelling: the algorithm will suggest and rank alternate surface forms which vary along these dimensions.

In future work we hope to deploy morph completion models in mobile devices, to support text entry. However, before we get there, we need to do proper user testing with members of the community and get their feedback on a polished demo of the project at this stage. Indeed, a limitation of this work is that we chose not to carry out thorough user testing in the Cree Community at this stage. It is natural for researchers to want to rush prototypes into the hands of prospective users, but this can lead to technology burnout among otherwise willing collaborators (Harrigan et al., 2019; Le Ferrand et al., 2022). We performed intrinsic evaluation by measuring the model's completion space to judge the feasibility of moving forward with the concept, and did self-testing to convince ourselves that the user experience is workable. That is the scope of this work.

Meaningful advances in language technology for low-resource, Indigenous, and/or endangered languages entails progress in our recognition and engagement with the context and use cases for such technologies at a community level. Morph-based autocompletion is designed to support text entry and word-building for morphologically rich languages. There are myriad factors which affect the usefulness of any approach in the real world. In our experience, connecting with real-world contexts leads to a better understanding of use-cases and problem constraints. This, in turn, fuels creativity and leads to better outcomes for the language communities we work with.

Acknowledgements

We are grateful to the Bininj people of northern Australia and the Plains Cree (*nêhiyawak*) communities, in particular those associated with the Maskwacîs Education Schools Commission (MESC) in Maskwacîs, Alberta, Canada, for the opportunity to work with them on language projects. Our thanks as well to the anonymous ACL reviewers and to Steven Bird for their feedback on earlier versions of this paper. We are also grateful to Miikka Silfverberg for making available his implementation for the transitional weighting of a HFST model. This research was supported in part by the Australian government through a PhD scholarship, and grants from the Australian Research Council and the Indigenous Language and Arts Program, and by a Partnership Grant (#895-2019-1012) from the Social Sciences and Humanities Research Council (SSHRC) of Canada.

References

Freda Ahenakew. 1987. *Stories of the House People told by Peter Vandall and Joe Douquette*. Winnipeg: University of Manitoba Press.

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *International Conference on Implementation and Application of Automata*, pages 11–23. Springer.

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. In *Proceedings of the 2016 Dene Languages Conference*, pages 51–69. Alaska Native Language Center, University of Alaska, Fairbanks.

Antti Arppe, Jordan Lachler, Trond Trosterud, Lene Antonsen, and Sjur N Moshagen. 2016. Basic language resource kits for endangered languages: A case study of Plains Cree. In *Proceedings of the 2016 CCURL Workshop. Collaboration and Computing for Under-Resourced Languages: Towards and Alliance for DigitalLanguage Diversity*, pages 1–9.

Antti Arppe, Katherine Schmirler, Atticus G Harrigan, and Arok Wolvengrey. 2020. A morphosyntactically tagged corpus for Plains Cree. In *Papers of the 49th Algonquian Conference (PAC49)*, volume 49, pages 1–16.

G Bear, M Fraser, I Calliou, M Wells, A Lafond, and R Longneck. 1992. *Kôhkominawak otâcimowiniwâwa/Our grandmothers' lives: As told in their own words, edited by Freda Ahenakew and H. Cristoph Wolfart*, volume 3. Regina: Canadian Plains Research Center.

Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.

Leonard Bloomfield. 1930. *Sacred Stories of the Sweet Grass Cree*. Department of Mines Bulletin.

Leonard Bloomfield. 1934. Plains Cree texts. *Publications of the American Ethnological Society*, 16:1–309.

Megan Bontogon, Antti Arppe, Lene Antonsen, Dorothy Thunder, and Jordan Lachler. 2018. Intelligent computer assisted language learning for nêhiyawêwin: an in-depth user-experience evaluation. *Canadian Modern Language Review*, 74(3):337–362.

Peter John Carroll. 1976. *Kunwinjku: a language of western Arnhem Land*. Canberra, ACT: The Australian National University.

Amy Dahlstrom. 2014. *Plains Cree Morphosyntax (RLE Linguistics F: World Linguistics)*. Routledge.

Steven Etherington and Narelle Etherington. 1998. *Kunwinjku Kunwok: A Short Introduction to Kunwinjku Language and Society: with Extra Notes on Gundjeihmi*. Gunbalanya: Kunwinjku Language Centre.

Nicholas Evans. 2003. *A Pan-Dialectal Grammar of Bininj Gun-Wok (Arnhem Land): Mayali, Kunwinjku and Kune*. Pacific Linguistics. Australian National University.

Kyle Gorman. 2016. Pynini: A Python library for weighted finite-state grammar compilation. In *Proceedings of the SIGFSM Workshop on Statistical NLP and Weighted Automata*, pages 75–80, Berlin, Germany. Association for Computational Linguistics.

Atticus Harrigan, Antti Arppe, and Timothy Mills. 2019. A preliminary Plains Cree speech synthesizer. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 64–73, Honolulu. Association for Computational Linguistics.

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(4):565–598.

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

Jim Kā-Nīpitēhtēw. 1998. ana kā-pimwēwēhahk okakēskihkēmowina/the counselling speeches of Jim kā-nīpitēhtēw.

William Lane and Steven Bird. 2019. Towards a robust morphological analyzer for Kunwinjku. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 1–9, Sydney, Australia. Australasian Language Technology Association.

William Lane and Steven Bird. 2020. Interactive word completion for morphologically complex languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4600–4611. International Committee on Computational Linguistics.

Eric Le Ferrand, Steven Bird, and Laurent Besacier. 2022. Learning from Failure: Technology for Data Capture in an Australian Aboriginal Community. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A Pirinen, and Miikka Silfverberg. 2013. HFST: a system for creating NLP tools. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 53–71. Springer.

Cecilia Masuskapoe. 2010. piko kīkway ē-nakacihtāt: kēkēk otācimowina ē-nēhiyawastēki mitoni ē-āh-itwēt māna cecila masuskapoe/there's nothing she can't do: Kēkēk's autobiography published in Cree. *Exactly as told by Cecilia Masuskapoe*.

Emma Minde. 1997. *Their Example Showed Me the Way: A Cree Woman's Life Shaped by Two Cultures*. University of Alberta Press.

Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M Tyers. 2014. Open-source infrastructures for collaborative work on under-resourced languages. *Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era*, pages 71–77.

Jean L Okimāsis. 2018. Cree: Language of the plains/nēhiyawēwin: paskwāwi-pīkiskwēwin.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Solomon Ratt. 2016. *Maci-nehiyawewin: Beginning Cree*. University of Regina Press Regina.

Aleksi Sahala, Miikka Silfverberg, Antti Arppe, and Krister Lindén. 2020. BabyFST - towards a finite-state based computational model of ancient babylonian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3886–3894, Marseille, France. European Language Resources Association.

Katherine Schmirler, Antti Arppe, Trond Trosterud, and Lene Antonsen. 2017. Computational modelling of Plains Cree syntax: A constraint grammar approach to verbs and arguments in a Plains Cree corpus. In *49th algonquian conference, Montreal, QC*.

Conor Snoek, Dorothy Thunder, Kaidi Loo, 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. Association for Computational Linguistics.

Sarah Whitecalf. 1983. *The Cree language is our identity: The LA Ronge Lectures of Sarah Whitecalf*. Univ. of Manitoba Press.

H. Christoph Wolfart. 1973. Plains Cree: A grammatical study. *Transactions of the American Philosophical Society*, 63(5):1–90.

H. Cristoph Wolfart. 2000. Introduction & notes. âh-âyîtaw isi ê-kî-kiskêyihtahkik maskihkiy/they knew both sides of medicine: Cree tales of curing and cursing told by Alice Ahenakew, ed. by H. Christoph Wolfart and Freda Ahenakew.

Arok Wolvengrey. 2001. *nêhiyawêwin itwêwina = Cree: Words*, bilingual edition edition. University of Regina Press, Regina.

Arok Wolvengrey. 2012. The verbal morphosyntax of Aspect-Tense-Modality in dialects of Cree. Paper presented at the 2nd International Conference on Functional Discourse Grammar, Universiteit Ghent, Ghent, Belgium.

Arok Elessar Wolvengrey. 2011. *Semantic and pragmatic functions in Plains Cree syntax*. Netherlands Graduate School of Linguistics.